משחקים ATARI®
ושעשועים



אטארי משחקים ושעשועים

מאת: הרב קול, טד קאהן, לן לינדסי
בשיתוף עם פאט קלילנד

איורים: סטיב אוליף
תרגום: רות לוין
עיצוב המהדורה העברית:
הדס בר-יוסף, דינה שהם

תוכן העניינים

פרק ה'

חלק שלישי

פרק ו'

פרק ז'

הקדמה

יש דרכים רבות ושונות לגשת למחשב אטארי הביתי שלך, ולא משנה אם הוא אטארי 400 או אטארי 800. אפשר לשחק אתו, ליצור תוכניות משלך ; וגם לחקור את לוח המקשים ואת המקשים המיוחדים לגרפיקה, קול וצבע הבנויים אל תוך המכשיר. דרך אחרת היא להעתיק תכניות שאחרים כתבו עבורך ולהריץ אותן.

בספר הזה נראה לך משחקים רבים שאפשר להעתיק ולשחק במחשב אטארי האישי שלך ; אפשר גם לכתוב תכניות ומשחקים משלך, ולעצב ולעטר את המשחקים כך שיראו יפים ומענינים על המסך.

עיטור משחקים, כך שהצגתם על המסך תהיה מעניינת, וכן שבמשחק יהיו הומור ואישיות, הוא רק חלק ממה שאפשר לכנות האישיות של התיכנות. לכל מחשב יש אישיות משלו, וכל אדם שמבלה בעבודה ליד המחשב מפתח סגנון ואישיות שמאפיינים את תכניותיו. יש בדרך כלל דרכים רבות לכתוב תכנית, ודרכים רבות להציג את התכנית על המסך. מחשבים ותיכנות הם דברים גמישים, לכן כדאי לך לתת לסגנון האישי שלך למצוא את ביטויו.

ולא צריך לפחוד משגיאות ; להתחיל לעבוד עם מחשב זה כמו ללמוד שפה חדשה : השגיאות יתרמו לשליטה הסופית שלך במחשב ובאפשרויות שלו.
חלק מספר זה מיועד למתחילים. כל פרק בחלק הראשון כולל הצגה של רעיונות ופקודות שבשמוש התכניות המוצגות בפרק. יחד עם זאת, בסוף כל פרק יש תכניות מתוחכמות שצריכות לעניין גם מתכנתים מנוסים.
ככל שנתקדם בספר, נסביר יותר מן המרכיבים של שפת המחשב שבהם משתמשות התכניות.

נדגים גם כמה תפיסות יסודיות כמו לולאות והסתעפויות שיאפשרו לך לפתח תכניות מורכבות. אפשר להשתמש בספר הזה כספר לימוד בסיסי, וגם כספר משחקים.
למעשה, אפשר להשתמש בחלק גדול של החומר בספר גם עבור מחשבים אישיים אחרים כדי ללמוד ולהשתלט על אוצר המלים והמושגים המרכזיים של שפת BASIC.

אטארי הוא סמל מסחרי רשום של חב' אטארי בע"מ (חברה השייכת ל"וורנר קומיוניקיישנס") והוא בשמוש ברשות.

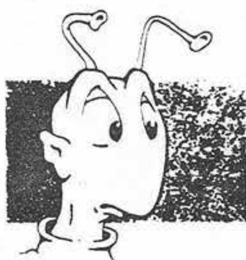
מחשב אטארי מתאים במיוחד ללימוד באמצעות משחקים,
משום שהוא צמח ממשחקים כמו: "PONG", והוא מאפשר
לתכנת משחקים מורכבים לשמוש עם JOYSTICKS ו-
PADDLES. ניתן גם לרכוש חבילות תוכנה של משחקים
למחשב שלך, וכן JOYSTICKS ו- PADDLES.
אבל אפשר גם, ככל שמתנסים יותר, לתכנת משחקים משלך
שגם הם ישתמשו בעזרים האלה - והמשחקים שלך יהיו בעלי
אתגר כמו אלו שקנית.

זו תמונה של JOYSTICK. בודאי שמת לב ששתי דמויות
נבוכות יושבות עליו, תהית האם לשחק "STAR RAIDERS"
או ליצור את המשחקים שלהן בעצמן.

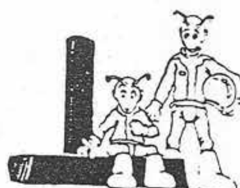
בעזרתו של סטיב אוליף, שעבד בחברת MARVEL (שפרושה
"קסם") ובהוצאות לאור אחרות של ספרים מצוירים,
הבאנו לעולם את הזוג הזה, הורה וילד (אנחנו לא
יודעים מה מינם, ואין לנו אפילו מושג איזה מין
מינים קיימים בחלק של החלל שממנו הם באים).

סטיב אומר, ששמו של האב הוא טא*רי, ושמה של הבת
הוא טא*רי הקטנה.

למרות מה שחושבים בדרך כלל על המצב הטכנולוגי
המתקדם והתחכום של יצורים מהחלל החיצון, טא*רי
והקטנה יודעים פחות על מחשבים ממה שיודע הקורא
המתחיל שלנו.


Ta*ri


Ta*ri Junior



דלתא - דרמה 2, הגלקסיה שממנה הם באים, יותר מתקדמת
משלנו מכמה בחינות. למשל, אף פעם לא היתה להם בעיה
להתקשר בחלל או בזמן. הדרמרים השקיעו את כל כוחם
בפיתוח כוחות פנימיים לתקשורת ורק עכשו הם מתחילים
לגלות את המחשב. טא*רי הקטנה ואביה החליטו להעביר את
עצמם לכדור הארץ ולחקור את עולם המחשבים.

PONG & STAR RAIDERS הם שמות מוגנים של אטארי.

אנחנו נשתמש בהם לאורך כל הספר כדי להסב את תשומת לבך למושגים חדשים, לחלקים חשובים מאד בתכניות, להסבר של תכניות ולמקומות בהם אפשר בקלות לשנות תכניות שלנו ולהכניס שנויים וחידושים משלך.

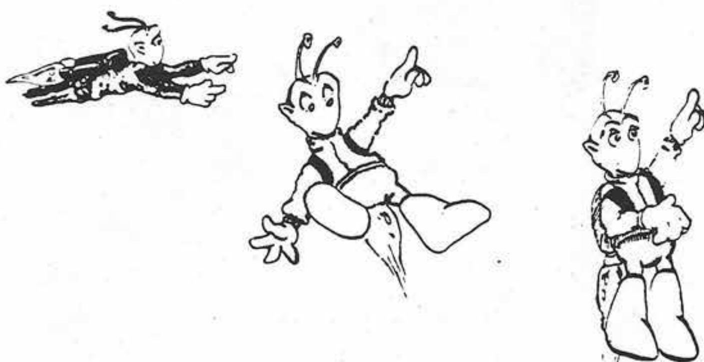


התמונה הזו תופיע בשולים של הדף, בכל פעם שנציג מילה חדשה ב BASIC



התמונה הזו תלווה אותנו כשנסביר מושגים, רעיונות בתוכנית או את מבנה המחשב.

יש חלקים מסוימים בתכניות שהם יותר חשובים מחלקים אחרים. לעתים קרובות, די קשה לאנשים שלומדים תכנות להבחין בין החלקים האלה לבין הוראות, הערות או פיתוחים לא חיוניים של התכניות. טא*רי הקטנה תציין נקודות מרכזיות במיוחד בתכניות בדרכים הבאות:



נוסף לנקודות חשובות, יש מקומות בתכניות מחשב שבהם צריכים להיות זהירים באופן מיוחד שלא לשגות. לפעמים רוצים להציג נתונים חדשים או לנוע קדימה ואחורה לחלקים שונים של התכנית. במקרים כאלה,

וגם באחרים, קל לשגות. אנחנו נציין בשבילך את אותם המקומות בהם צריך להזהר באופן מיוחד מפני שגיאות. הנה, כך נציין זאת:





יש גם מקומות רבים שבהם יהיה ברצונך אולי לשנות תוכנית או להכניס לתכנית רעיון חדש. אנחנו נציין מקומות כאלה, ע"י טא*רי הקטנה.



זאת התמונה שתופיע במקומות שבהם אנו חוזרים על מושגים ומילים ב BASIC

אנו מקוים שטא*רי והקטנה יעזרו לך למצוא את דרכך בין כל התכניות שלנו ויאפשרו לך לכתוב תכניות בנוחיות וככל שידריך אותך הדמיון.

הנה עוד כמה נקודות שתוכלנה לעזור לך :

יש כמה מקשים מיוחדים על לוח המקשים. דוגמאות אחדות הן (RETURN), (SHIFT), (BREAK), ו-(CLEAR). אנחנו נציין את המקשים המיוחדים האלה באותיות גדולות, ובתוך סוגריים.

לפעמים חשוב להשאיר רווח כחלק מן התכנית; כדי ליצור רווח, פשוט נלחץ על מקש הרווחים הגדול שבתחתית לוח המקשים. רבוע קטן מלא ■ יציין בשבילנו, בהדפסים של התכניות בפרק א', מקומות בהם דרושים רווחים בתוך התכנית. כשנגיע לפרק ב' כבר יהיה לך קל לזהות את המקומות האלה בלי שנזכיר לך באופן מיוחד.



המגף של טארי הקטנה ישמש אותנו לציין הערות בתחתית העמוד. מאחר שכוכבית (*) משמשת בתכנות לציון סימן כפל, הרי שהמגף של טא*רי ימנע מאתנו בלבול מיותר.

אנו מציגים בספר סוגים שונים של משחקים. בחלק הראשון נציג כמה משחקים בתור התחלה, וכן את המירוץ המופלא בין הצב והארנבת. נכלול גם קטע על משחקים שיכולים לעזור לילדים ללמוד קצת חשבון. גם הלומד המתחיל, או המתחילה, יכולים לבנות מבחן או תרגילים בחשבון, ויכולים ללמד את הבן או את האח או האחות הקטנים, לעשות אותו דבר.

בחלק השני נציג גם משחקי ניחוש מילים. חלקם נראים אולי קצת טפשיים, אבל תענוג לשחק אותם, וגם כמה משחקי מספרים. דרך המשחקים האלה המחשב שלך ישמש אותך כשעשוע במסיבות לילדים ולמבוגרים כאחד. יהיו בהם משחקים לניחוש **עתידות**, **חידות**, בדיחות ומשחקי מסתורין.

החלק השלישי ידון במרכיבים המיוחדים של המחשב האישי של אטארי שמטפלים בגרפיקה, קול וצבע. נלמד לצייר גרפים, לצייר דמויות הנפשה, לחבר שירים, להוסיף קול וצבע למשחקים שלך, וליצור משחקים ופעילויות שישלבו את כל הצורות השונות האלה של פעולת המחשב.

הנספחים בסוף הספר מכילים מספר טבלאות, כרטיסים ושרטוטים גרפיים שישמשו אותך במשחקים עם המחשב שלך. מילון השגיאות והטעויות יעזור לך לטפל בשגיאות תכנות שהמחשב יכול להצביע עליהן. מאחר שאנו נמצאים רק בשלב ההתחלה של התכנות עכשיו, חשוב להבין ששגיאות בתכנות הן דבר כל כך רגיל ונפוץ (גם אצל מתכנתים מנוסים), עד כי

המכשיר מכיל תכניות שנבנו לתוכו במיוחד שאומרות לך מתי בצעת שגיאה נפוצה וגם נותנות לך הנחיות איך לתקן את השגיאה.

איזה ציוד דרוש כדי לשחק במשחקים שבספר: ובכן, דרוש מחשב אישי אטארי 400 או אטארי 800. על המחשב להיות מצויד בשנאי ובכרטיס התחברות לטלביזיה (שני אלה הם ציוד סטנדרטי למחשבי אטארי), וכן ב"קסטה" ATARI BASIC. מחשב אטארי יכול להתחבר לכל מסך טלביזיה, שחור־לבן או צבעוני. אבל, מאחר שרבים מהמשחקים בספר משתמשים ביכולת

המחשב ליצור צבע, הרי שהמשחקים יראו טוב יותר אם המחשב יתחבר לטלביזיה צבעונית.

חלק מן התכניות בספר זקוקות למינימום 16K RAM זכרון, אבל אפשר לתכנת את רוב המשחקים גם עם 8K זכרון.

אנו גם ממליצים שיהיה בידך עותק של ATARI 400/800 BASIC REFERENCE MANUAL וכן עותק של הספר ATARI BASIC. כדי לשמור את המשחקים כך שאפשר יהיה לשחק בהם שוב בעתיד, דרוש לרש קסטות ATARI410 (אשר משתמש בקסטות רגילות להקלטה) או כונן דיסקטים ATARI 810 עם 810 MASTER DISKETTE או עם MASTER DISKETTE II 810. (אטארי 810 משתמש בדיסקטים בקוטר "5 1/4).

כמה מן המשחקים בספר דורשים שמוש במוט הבקרה. ATARI CX40, או ATARI CX30 PADDLE. העזרים האלה מוסיפים הרבה ל"משחקיות" של משחקים רבים, והם יהיו השקעה נבונה בהעלאת ערך השעשוע והבלוי של המערכת.

אין הכרח לקרוא את הספר באופן רציף. אם יש לך נסיון במחשבים, אפשר להתחיל בכל מקום שרוצים ולהשתמש באותם פרקים שמהוים אתגר עבורך. לחדשים בעניני מחשב, עדיף להתחיל בחלק הראשון ואז להמשיך בתוך הספר ומחוצה לו ולחקור את סוגי המשחקים השונים שהמחשב יכול להריץ. אנו ממליצים לשחק משחקים רבים ככל האפשר, מכיון שבאמצעותם ניתן להכיר טוב יותר את המכשיר. אולם, אנו מקוים שיותר משתשחקו במחשב אטארי תשחקו אתו ותשתלטו עליו. בסופו של דבר האדם הוא השכל שמאחורי המחשב שלך. ככל שיגדל הידע וככל שתגבר ההתנסות, כך גם יהיו הדברים שאפשר לעשות עם מחשב אטארי מענינים יותר. המחשב יכול להיות שמושי, הוא יכול לספק בילוי, והוא יכול אפילו להיות אנושי אם האדם יעשה אותו כזה.

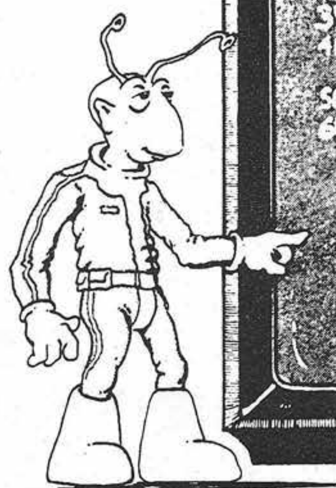
הבעת תודה

המחברים רוצים להודות לאנשים הבאים, שאיפשרו את הוצאת הספר הזה לאור: ניקי הרדין, העורך שלנו בבית ההוצאה לאור, רסטון; ובארי וינסנט, שסייע בעריכת כמה מהתכניות.
אנו מבקשים להביע תודה מיוחדת לדייל דישרון על בדיקת כל המשחקים, ועריכת רוב התכניות בספר; וכן לסטיב אוליף, אשר צייר את הדמויות המקסימות של טא*רי וחבריה.
תודות רבות גם לג'ודי ופרונה על סבלנותן; ולראינון, ילד מקסים בן ארבע אשר בדק כמה משחקים עבור לן לינדסי.
ואחרון אחרון חביב, תודות לפיטר רוזנטל ומחלקת המחשבים הביתיים של אטארי, על שסיפקו למחברים.מחשבי אטארי 800 לעבודתם.

חלק ראשון

הבה נלמד
על המחשב הביתי אטארי
דרך משחקים

פרק א'



איך מתקשרים עם המחשב



הדבר הראשון שעליך לזכור לגבי המחשב שלך הוא שהמחשב מקבל פקודות ומבצע הנחיות. האדם הוא זה שאומר לו מה לעשות. אם ההנחיות שלך מעניינות, המחשב יעשה דברים שקשה מאד לעשותם עם נייר ועפרון. אם ההנחיות שלך שגויות, המחשב עלול לקבל התמוטטות עצבים ולבצע טעויות הדורשות תיקון. יש אנשים שחולמים על בנית מחשב שיתקן את עצמו, אבל בינתיים זהו רק חלום.

כדי לתת למחשב הוראות, צריכים להתקשר עם המכשיר באמצעות שפה משותפת. שפה משותפת כזאת למחשב הזה ולאנשים שמשתמשים במכשיר, נקראת BASIC (בייסיק). למוד שפת BASIC של אטארי לא יותר קשה מלימוד כמה מלים חדשות בשפה זרה, או מלימוד רשימת מלים להכתבה בבית הספר. בספר הזה לא נשתמש בכל אוצר המלים של ATARI BASIC. בעצם, נשתמש בפחות משלושים מילים. כוחו של המחשב הוא בכך שגם באמצעות אוצר מלים קטן ביותר הוא יכול לבצע דברים מורכבים ומתוחכמים.



כששולטים באוצר המלים שנציג כאן, מתחילים בעצם לדבר ולחשוב בשפת BASIC. שליטה באוצר המלים היא דרך ללמוד כיצד פועל המחשב; וכשמבינים זאת, מגלים שאפשר להתמודד עם בעיות יותר ויותר מורכבות בתכנות.



מחשב אטארי שלך הוא מערכת גמישה שיכולה להתרחב ולהכיל כמה חלקים. החלק החשוב ביותר הוא לוח המקשים ותיבת המחשב, שמכילה את יחידת העבוד המרכזית ואת הזכרון של המחשב. ההתקשרות שלך עם המחשב נעשית דרך לוח המקשים ותיבת המחשב.

לוח המקשים של המחשב דומה ללוח המקשים של מכונת כתיבה ויש לו תכונות רבות הדומות לאלו של מכונת כתיבה רגילה.





יחד עם זאת יש לו כמה תכונות מיוחדות, – שאותן נציג במהלך הספר. תכונה מיוחדת אחת היא שאף פעם אי אפשר להחליף בין מספרים ואותיות. במכונת כתיבה רגילה אפשר להחליף בחפשיות בין המספר 1 לבין האות L, ובין הספרה 0 לבין האות O. חילופין כאלה יבלבלו את המחשב.

לכן חשוב לזכור: לגבי רוב תכניות המחשב, ספרות ואותיות הם שני תחומים נפרדים לחלוטין.

ועכשיו נעבור ל BASIC. שתי המלים הראשונות שנלמד הן RUN ו- LIST.△

פקודת RUN מורה למחשב לבצע את התכנית שנמצאת <u>בזכרון</u>. התכנית יכולה להיות משחק, או טבלה לחישובי מס או דמויות בסרט הנפשה. התכנית שנמצאת ברגע זה בזכרון של המחשב תבוצע עם תקתוק RUN ולחיצה על המקש (RETURN). הדלקת המחשב ברגע זה, תקתוק RUN ולחיצה על המקש (RETURN) לא יגרמו לשום פעילות משום שאין כל תכנית בזכרון המחשב. בנקודה זו של ההתחלה, עליך לשים משהו בתוך הזכרון.

כשמדפיסים LIST ולוחצים על המקש (RETURN) ידפיס המחשב על מסך הטלביזיה שלך את תוכן התכנית שיושבת אצלו בזכרון. אם נתקתק עכשיו LIST לא יופיע שום דבר מכיון שאין כל תכנית בזכרון. המסך פשוט יראה READY (מוכן), כלומר יציין שאפשר להתחיל לתכנת ברגע שהמתכנת מוכן לכך.

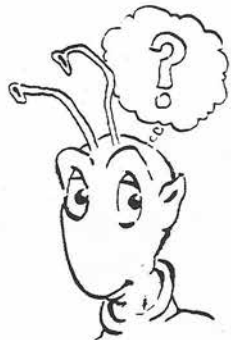


על מנת שאפשר יהיה להכניס תכנית לזכרון של המחשב עליך לתת תוית, או מספר לכל שורה שמופיעה בתכנית. תוית זו נקראת <u>מספר השורה</u>. אפשר להתחיל בכל מספר שרוצים (בתנאי שזהו מספר שלם, בין 0-32767). בהמשך התכנית מותר שיהיו מספרי שורות שלא בסדר רציף. אבל המחשב יסדר מחדש את מספרי השורות כך שיהיו בסדר עולה. <u>חשוב לזכור:</u> לא חשוב באיזה סדר יכתבו מספרי השורות בתכנית – המחשב יסדר אותם תמיד בסדר עולה – מהמספר הקטן ביותר למספר הגדול ביותר, ויבצע את התוכנית לפי הסדר הזה.

△19 כל פקודות ה BASIC במחשב ATARI נכתבות תמיד באותיות גדולות.

למשל, אם כתבת את מספרי השורות בסדר הזה
100
50
10
250
המחשב יסדר אותם מחדש ויבצע אותם בסדר הבא:
10
50
100
250

אם לא יופיע מספר שורה לפקודת BASIC, ינסה המחשב לבצע את פקודת BASIC מיד, ולא יאחסן את הפקודה בזכרון. למשל, בפקודות RUN ו- LIST לא כותבים מספר שורה: הן "מבוצעות" מיד כשגומרים את השורה ע"י לחיצה על (RETURN).



העובדה שאפשר לנסות את רוב משפטי BASIC במערכת ATARI במצב הזה של "בצוע מידי" היא דרך נוחה ללמוד פקודות חדשות ב- BASIC. זוהי גם דרך שימושית מאד לגילוי שגיאות תכנות, כפי שנראה בהמשך. אולם, אם רוצים שהמחשב יזכור תכנית BASIC (אפילו אם זו תכנית בת שורה אחת), מוכרחים לציין מספר שורה לכל שורה.



ברגע שהכנסנו תכנית לזכרון של המחשב, היא תשאר שם, אלא אם כן: 1) ננתק את המחשב, 2) נדפיס את הפקודה NEW ונלחץ על (RETURN).

חשוב לזכור שתי נקודות אלה. אנשים שאינם רגילים למחשב אישי שוכחים לפעמים שניתוק המכשיר מוחק את הזכרון. אם נמצאים באמצע תכנית ורוצים לשמור אותה, צריך להשתמש ברשם הקסטות ATARI 410 או בכונן הדיסקטים ATARI 810 (לא לשכוח לבדוק את ההנחיות להפעלתם במדריך למשתמש של ATARI).
לבסוף, חשוב לזכור שעם פקודת NEW ניתן לנקות את הזכרון של המחשב ולכתוב תכנית חדשה.



PRINT היא פקודה שאומרת למחשב להדפיס משהו על פני מסך הטלביזיה. פשוט , אבל לא לגמרי. אם ברצונך שהמחשב ידפיס את שמך, הפקודה המתאימה היא:
PRINT כששמך יופיע בין מרכאות אחרי הפקודה, ואז ללחוץ על (RETURN).

PRINT "NAME" (RETURN)

השם צריך לבוא באנגלית בין המרכאות. המרכאות בפקודת PRINT חשובות, כדי שהמחשב ידפיס בדיוק את מה שכתבת. הבה ננסה להורות למחשב להדפיס את המשפט הבא:

I KNOW HOW TO TYPE THIS SENTENCE
(שפירושו "אני יודע להדפיס את המשפט הזה").
אחרי תקתוק המשפט :
PRINT "I KNOW HOW TO TYPE THIS SENTENCE"
חשוב לזכור ללחוץ על מקש (RETURN). המחשב יבצע את הפקודה שלך אחרי הלחיצה על (RETURN). האם זכרת להשתמש במרכאות?

פקודת PRINT פועלת באופן שונה כאשר לא משתמשים במרכאות. כשאין מרכאות, הוראת PRINT אומרת למחשב לפעול כמו מכונת חישוב.

PRINT 3+3
תגרום למכשיר להדפיס את התוצאה, 6.

הבה נראה מה קורה כשניתן למכשיר את פקודת PRINT הבאה:



```
PRINT "23+45"   (RETURN)
PRINT 23+45     (RETURN)
PRINT 45*45     (RETURN)
PRINT "45*45"   (RETURN)
```

אם ננסה עכשיו לתת פקודת PRINT למלים או אותיות ונשכח את המרכאות, כל מה שנקבל הוא 0 (לא האות O), כי המחשב נותן ערך של 0 אלא אם כך אומרים לו אחרת.



עכשיו ננסה:

```
PRINT CAT     (RETURN)
PRINT "CAT"   (RETURN)
```

וכעת ניתן למכשיר להדפיס

```
PRINT "I AM JUST A MACHINE"
```

(שפירושו "אני רק מכונה"). לחיצה על (RETURN) תיתן את המשפט שכתבת. מה קורה אם נותנים למכשיר פקודת ?RUN

כלום! - מפני שלא נתת תוית (מספר שורה) לשורה עם הוראת PRINT, ולכן לא נכנסה השורה לזכרון המחשב. הנה ההוראה בצירוף מספר השורה:



```
10 PRINT "I AM NOTHING BUT A MACHINE"
```



ועכשיו נראה: בבקשה ללחוץ על (RETURN), לתקתק RUN ושוב ללחוץ (RETURN): המכשיר יבצע את ההוראה שלך. בעצם בכל פעם שהמחשב יקבל פקודת RUN ואחריה (RETURN), הוא ידפיס את השורה. המחשב זוכר את ההוראות שלך ומבצע אותן: זוהי תמצית התכנות.

עם תקתוק LIST ולחיצה על (RETURN) התכנית בת השורה שכתבת זה עתה תודפס על המסך;
חשוב לזכור, RUN גורמת לתכנית לפעול, או "לרוץ" ו LIST מראה לך את התכנית.



וכעת למלה חדשה ומושג חדש. המילה היא GOTO. GOTO אומרת למחשב ללכת לשורה מסוימת בתכנית ולבצע מה שאומרת השורה, ואז להמשיך בשורה הבאה. GOTO יכולה להתחיל לולאה, שהיא אחד הדברים רבי העצמה שמחשב יכול לעשות. נסתכל בתכנית הבאה:

```
10 PRINT "I AM NOTHING BUT A MACHINE"
20   GOTO  10
```

תכנית פשוטה זו אומרת להדפיס את המשפט "I AM NOTHING BUT A MACHINE" ואז ללכת לשורה 20. אבל שורה 20 אומרת לחזור לשורה 10 ולבצע אותה שוב.



כלומר, התכנית נעה חזרה בצורת לולאה בגלל פקודת GOTO. ניתן פקודת RUN. מה קורה? המשפט חוזר על עצמו שוב ושוב וחוזר חלילה. התכנית תרוץ כל זמן שהמחשב שלך או מסך הטלביזיה יפעלו, אלא אם כן נפסיק אותה. יצרת כאן לולאה אינסופית שאפשר להפסיקה פשוט ע"י לחיצה על מקש (BREAK). כדי להתחיל את התכנית מחדש ממקום שהפסיקה, צריך לתקתק CONT (מן המלה CONTINUE, להמשיך).

כעת נעשה את הלולאה הפשוטה הזו קצת יותר מענינת. ע"י תוספת ; (נקודה פסיק) בסוף הוראת PRINT, ידפיס מחשב אטארי את המשפטים בזה אחר זה, במקום בכל שורה בנפרד. כך יתמלאו כל פני המסך. אבל, המשפטים יודפסו ביחד באופן הבא:




```
I AM NOTHING BUT A MACHINEI AM NOTHING
BUT A MACHINEI AM NOTHING BUT A MACHI
NEI AM NOTHING BUT A MACHINEI AM NOTHI
```

הוספת רווח אחרי המלה האחרונה במשפט והצבת המרכאות באופן הבא:

```
10 PRINT "I AM NOTHING BUT A MACHINE■";
```

יתנו את המשפט בצורה יותר קריאה והמשפט יחזור על עצמו:

```
I AM NOTHING BUT A MACHINE I AM NOTHIN
G BUT A MACHINE I AM NOTHING BUT A MAC
HINE I AM NOTHING BUT A MACHINE I AM N
```



הסיבה לכך היא, שכל דבר הנמצא בין המרכאות (כולל רווחים) ישוחזר ע"י המחשב. לפעמים מרגישים מאד מדוכאים ומרגישים שאף אחד לא זוכר את שמנו. כדאי אז לנסות את התכנית הזו שתמלא את כל המסך, ותזכיר לכולם את קיומנו :

```
10 PRINT "YOUR NAME■";
20 GOTO 10
```



והנה לולאה קצת יותר מורכבת: לפני התקתוק לתוך המכשיר כדאי לקרוא את התכנית ולנסות להבין מה היא תעשה על המסך. כדאי להתרגל לדמיין לעצמך את התכניות ולנחש כיצד הן תיראנה על המסך אחרי הריצה. בשלב יותר מאוחר, יתגלה הרגל זה כשמושי מאד לתקון תכניות מסובכות שלא עושות בדיוק מה שרצית שתעשינה.

```
10 PRINT "I DON'T LIKE SPINACH■";
20 PRINT "BUT POPEYE DOES■";
30 GOTO 10
```

 ב BASIC של ATARI, המחשב שלך מציג 38 תוים בכל שורה לפני שהוא עובר לשורה הבאה על המסך.

ננסה עוד שנוי פשוט. נחליף את ; שבסוף כל שורה
ב , (פסיק), ואז נריץ את התכנית. התכנית שלך
צריכה להראות כך:

```
10 PRINT "I DON'T LIKE SPINACH■",
20 PRINT "BUT POPEYE DOES■",
30 GOTO 10
```

בדומה ל ; גם הפסיק הוא אמצעי לציון רווח.
הוא מזיז את ההדפסה לעמודה (TAB) הבאה במחשב
שלך לפני שהמחשב עובר לשורה הבאה. ננסה
להשתמש ב ; בשורה אחת ובפסיק בשורה הבאה.
מהו השינוי שחל בהדפסה על המסך?



לומדים!

ואלה המושגים שצריכים להיות ברורים לך בשלב זה:
מילים ב BASIC, מקשים מיוחדים וסימני פיסוק
(סימנים כדוגמת המרכאות שאומרים למחשב כיצד
לבצע את הפקודה).

מושגים

☆ **תכנית**- סדרה של הוראות שמכניסים לזכרון המחשב ואשר
קובעות מה על המחשב לעשות.

☆ **מספר שורה**- מספר שלם, חיובי (קטן מ 32768)
שפותח כל "שורה לוגית" בתכנית.
המספרים האלה מורים למחשב לאחסן את המידע שבשורה
הזו בזכרון. באופן בלתי תלוי בסדר שבו נתקתק את
השורות בתכנית, המחשב יסדר אותן מן המספר הקטן
לגדול. כשהמחשב מבצע את התכנית, הוא יתחיל במספר
השורה הנמוך ביותר וירוץ דרך ההוראות בסדר עולה.
שורה הגיונית (לוגית) - מה שמדפיסים מן הספרה
הראשונה של מספר השורה עד הלחיצה על מקש
(RETURN). שורה הגיונית יכולה לתפוס שתי שורות
או יותר על המסך, אם היא מכילה יותר מ 38 תוים.

☆ **לולאה**- לולאה מתקיימת בתכנית כאשר בשורה אחת
מורים למחשב לחזור לשורה קודמת ולבצע פקודה או
סדרת פקודות. לולאה אינסופית מתרחשת כאשר סדרה
של פקודות מבוצעת שוב ושוב ללא סוף מפני
שהתכנית לא מכילה הוראה להפסקת הלולאה בנקודה
מסוימת.

מלים ב־ BASIC ☞

☆ RUN - הוראה לבצע תכנית הנמצאת בזכרון המחשב.

☆ LIST - הוראה להדפיס את כל שורות התכנית הנמצאת בזכרון המחשב.

☆ READY - מלה זו מופיעה על המסך כדי לציין שהמחשב מוכן לקבל תכנית, לתת הדפס שלה עם פקודת LIST או להריץ אותה אחרי שיקבל פקודת RUN.

☆ PRINT - פקודה זו פועלת בשתי דרכים; 1) בלי מרכאות, PRINT שאחריה בטוי חשבוני הופכת את המחשב למכונת חישוב. עם לחיצה על (RETURN) ידפיס המחשב את התשובה לבעיה החשבונית. 2) PRINT שאחריו כל ביטוי שהוא בתוך מרכאות מורה למחשב להדפיס בדיוק את כל מה שנמצא בתוך המרכאות.

☆ NEW - פקודה זו ואחריה (RETURN) מנקה את הזכרון של המחשב.

☆ GOTO - הוראה זו בשורה מסוימת מורה למחשב ללכת לשורה המצוינת אחרי ה GOTO ולבצע את הפקודה הכלולה בה. כך, GOTO 100 פירושו שהמחשב יבצע עכשיו את שורה 100 וימשיך מכאן. ע"י שמוש ב GOTO אפשר לומר למחשב לדלג קדימה ואחורה בתכנית.

התכנית

```
10 PRINT "X"
20 PRINT 2+2
30 GOTO 10
```

מורה למחשב לדלג אחורה ולבצע את שורה 10.

☆ CONT - ההסבר מופיע תחת הערך (BREAK).

מקשים מיוחדים ☞

☆ (RETURN) - לוחצים על מקש (RETURN) כשמסיימים שורה ורוצים להכניס אותה לזכרון ולהמשיך בשורה הבאה, או כשנותנים הוראה כמו LIST או RUN ורוצים שהמחשב יתחיל לבצע את ההוראות שלך.

☆ (BREAK) - מקש (BREAK) מפסיק תכנית או הדפסה של תכנית ברגע שלוחצים עליו. אם ברצונך להמשיך את הרצת התכנית, צריך לתקתק CONT ואז ללחוץ על (RETURN).

סימני פיסוק ☞

מרכאות - משתמשים ב "-" בפקודת PRINT כדי להורות למחשב להדפיס בדיוק את מה שנמצא בתוך המרכאות.

; - מורה למחשב להדפיס את הדברים בשורה רצופה, זה אחר זה. בפקודת PRINT

בלי ; נקבל

```
I AM THINKING
I AM THINKING
I AM THINKING
```

ועם ; נקבל

```
I AM THINKING I AM THINKING I AM THINK
ING I AM THINKING I AM THINKING I AM T
```

נסי,או נסה, לכתוב כמה תכניות משלך ולהשתמש במושגים ובמלים שלמדת עד עכשו. הנה כמה תכניות פשוטות שיעזרו לך להתחיל. רצוי לנסות לתאר לעצמך מה יעשו כשירוצו לפני כתיבתן אל המחשב והרצתן למעשה. אחר כך כדאי לשנות את התכנית, ולשחק עם המלים הפשוטות ב BASIC שהצגנו עד עתה.

חשוב לשים לב ששנויים קטנים בתכנית יכולים לגרום להבדלים גדולים בביצוע של המחשב למעשה!

תכנית 1A

```
10 PRINT "I DON'T WANT TO■"
20 PRINT "STAY■"
30 GOTO 10
```



תכנית 1B

```
10 PRINT "I DON'T WANT TO■"
20 PRINT "STAY■";
30 GOTO 10
```



תכנית 2A

```
10 PRINT "*** - - ***"
20 PRINT "** - - ** - - **"
30 GOTO 10
```



תכנית 2B

```
10 PRINT "*** - - ***";
20 PRINT "** - - ** - - **";
30 GOTO 10
```



כדאי לנסות גם לולאות GOTO פשוטות המשתמשות במספרים, סמנים ורווחים. כדי לבדוק את הצורה שנוצרה על המסך, לוחצים על (BREAK) והצורה תוקפא על המסך.

זו צורה קצת יותר מסובכת, לעיונך:

```
10 PRINT "1111111";
20 PRINT "0000000";
30 PRINT "@@@@@@@";
40 GOTO 10
```

כדאי לך לבנות צורות משלך, לנסות לדמיין אותן, ולראות אם הן באמת משוחזרות על המסך כפי שחשבת.



שגיאות - וכיצד מתקנים אותן ☞

לא צריך לפחוד משגיאות בשעה שלומדים לתכנת את המחשב. ככל שההתנסות בצרופים שונים של פקודות ומספרים וסימנים תגדל, כך יעשה לוח המקשים ברור יותר והתכנות יעשה יותר קל.

גם המתכנתים המתקדמים ביותר שוגים. לפעמים הטעויות פשוטות, כמו הדפסת PRIMT במקום PRINT או GOSO במקום GOTO, ולעתים הן יותר מורכבות וכוללות מתן הוראות למחשב שאין הוא יכול לבצע, כמו:

```
10 PRINT "HELP"
20 GOTO 15
```

(המחשב אינו יכול לעשות זאת, משום ששורה 15 איננה קיימת).

המתכננים של אטארי בנו אל תוך המחשב וכן אל תוך שפת BASIC של המחשב האישי שלך כמה מרכיבים שתפקידם לסייע לך להבין את הטעויות ולתקן אותן. מרכיב כזה הוא **מגלה טעויות** שמראה לך היכן טעית ובמקרים מסוימים אומר לך מהו סוג השגיאה שעשית.

המרכיב השני הוא **אמצעי עריכה** שמאפשר לך לתקן בקלות את השגיאות הללו.
על מנת להבין את שני המרכיבים האלה של המחשב האישי אטארי, הבה נבצע כמה שגיאות במתכוון ונראה מה קורה. נתחיל בשגיאת כתיב קטנה ונאמר למכשיר:

PRIMT "TOO BAD"

ועכשיו (RETURN). המכשיר יענה:

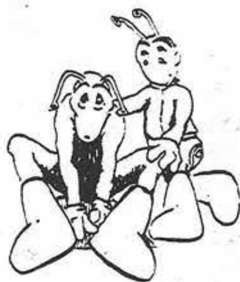
ERROR - PRIMT ["] TOO BAD"

ריבוע קטן מופיע מסביב למרכאות הפותחות.
זה מציין שהמכשיר לא יכול לבצע מה שפקדת עליו
לעשות, ועליך לחזור ולחפש את השגיאה בפקודה
(במקרה זה - הדפסת PRIMT במקום PRINT).

כעת **ננקה את המסך** ונבצע שגיאה קצת יותר מורכבת.
כדי לנקות את המסך בלי למחוק את הזכרון כפי שהדבר
נעשה כשמדפיסים NEW ולוחצים (RETURN), צריך
ללחוץ על (CLEAR) תוך כדי לחיצה על אחד ממקשי
(SHIFT). **הנתב** יופיע בפינה השמאלית העליונה
של המסך. הנתב הוא הרבוע הקטן שתמיד מציין
עבורך באיזה מקום ידפיס המחשב את האות או התו
הבא שיתקבל דרך לוח המקשים.

ננסה את התכנית הזו, המכילה שגיאה. (קודם
צריך לודא שנקית את הזכרון ע"י תקתוק NEW).

```
10 PRINT "ANOTHER ERROR"
20 GOTO 11
```



האם לחצת על (RETURN) אחרי כל שורה?
פקודת RUN ואחריה שוב (RETURN) יריצו את התכנית.
המחשב לא יוכל לבצע את ההוראה שלך
וידפיס שני דברים.ראשית: ANOTHER ERROR

תוך כדי ביצוע שורה 10. אח"כ ידפיס :

נא לא להבהל! פירוש הדבר

ERROR - 12 AT LINE 20

הוא,שבשורה 20, התיחסת לשורה שאינה קיימת בתכנית.
שגיאה 12, כפי שאפשר לראות ברשימת הודעות
השגיאה המפורטת במדריך למשתמש של ATARI
(ATARI BASIC REFERENCE MANUAL)
היא רמז עבורך שהשורה שאליה שלחת את המחשב ע"י
פקודת GOTO לא נמצאה ע"י המחשב בתכנית.
תיקון בתכנית כך שתהיה:

```
10  PRINT "ANOTHER ERROR"
20  GOTO 10
```

יגרום לכך שהשגיאה תעלם והתכנית תרוץ. **הודעות
השגיאה** עוזרות לך לגלות את התקלה. בנספח א יש
מידע על שגיאות והודעות שגיאה: כיצד תנפה את

השגיאות מתכניתך" פרטנו והסברנו את הודעות
השגיאה הנפוצות ביותר והמדריך למשתמש של
ATARI BASIC מכיל הסברים גם לשגיאות אחרות.

לא צריך להרגיש לא נוח בגלל טעות אחת או שתים;
לתשומת לבך: למעלה מ 50 הודעות שגיאה נבנו אל
תוך המחשב.גם כשהמתכנת כבר מנוסה ביותר,
תמיד אפשר לצפות שתהיינה שגיאות וקצת עזרה
לא תזיק.

נחזור עכשיו לתכנית עם המקשים.
ERROR - 12 AT LINE 20
לחיצה על (SHIFT)ו- (CLEAR) תנקה את המסך,
ופקודת LIST ואחריה (RETURN) יחזירו את
התכנית אל המסך וזה מה שיופיע:

```
10 PRINT "ANOTHER ERROR"
20 GOTO 11
```

נתבונן בלוח המקשים של המחשב. יש בו כמה מקשים
שמאפשרים לך לשנות תכנית, להוסיף או להוציא חלקים
מאמצע התוכנית ולתקן שגיאות. מקשים אלה נקראים
מקשי עריכה והם מבוקרים על ידי המקש הנמצא ממש
מעל המקש (SHIFT) השמאלי. כתוב עליו (CTRL)
הלקוח מן המלה CONTROL לבקר, לפקח.

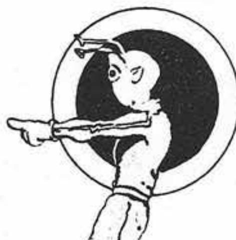




משתמשים במקשי הבקרה, ביחד עם המקשים האחרים,
כדי להזיז את הנתב (הרבוע על מסך הטלביזיה שמציין
היכן ידפיס המחשב את האות או התו הבא שנתקתק).
בימין לוח המקשים יש ששה מקשים מיוחדים לעריכה:
משתמשים בהם ביחד עם המקש (CTRL) שבשמאל לוח המקשים.

בימין השורות השניה והשלישית בלוח המקשים יש
ארבעה חיצים המציינים את הכוון שבו הם מזיזים את
הנתב. הבה ננסה משהו עכשו: נפעיל את הנתב ע"י
לחיצה והחזקה של מקש (CTRL) ובאותו זמן לחיצה על [→].
הנתב ינוע ימינה. עכשו לחץ על (CTRL) ועל [↑].
הנתב ינוע כלפי מעלה. כדאי להתנסות קצת
בהפעלת הנתב כדי לקבל את התחושה ואח'כ להביא
אותו לספרה 1 השניה במספר 11 שבתכנית:

```
10 PRINT "ANOTHER ERROR"
20 GOTO 11
```



כל מה שעליך לעשות עכשיו הוא לתקתק 0 וללחוץ על
(RETURN). זה יתקן את התכנית והיא תראה כך:

```
10 PRINT "ANOTHER ERROR"
20 GOTO 10
```

לחיצה על (SHIFT) ו- (CLEAR) תנקה את המסך,
ואז פקודות LIST ו-RUN לתכנית:השגיאה נעלמה.



אחת התכונות המיוחדות של לוח המקשים של מחשב
ATARI היא שלכל מקש יש מצב של "חזרה מהירה". אם
נלחץ ונחזיק את מקש x, לדוגמה, נקבל x-ים
חוזרים על עצמם על פני כל המסך. אפשר לנצל תכונה זו
כדי לזרז את העריכה עם מקשי העריכה המיוחדים.
למשל, ננסה את הדוגמה הבאה:

צריך ללחוץ ולהחזיק את מקש (CTRL) וביד השניה
ללחוץ **ולהחזיק** את מקש [→], זו דרך מהירה למקם
מחדש את הנתב בתוך אותה שורה. אותה פעולה עם [↑]
היא קיצור דרך למיקום הנתב בשורה אחרת.

יש עוד שני מקשי עריכה מיוחדים לשמוש עם (CTRL).
אלה הם (INSERT) ו- (DELETE) הנמצאים בצד הימני
העליון של לוח המקשים. באמצעות (INSERT) אפשר
להכניס, להוסיף סימנים ורווחים לתכנית וע"י
(DELETE) אפשר לסלק אותם מן התכנית.
נתחיל עם התכנית המתוקנת:

```
10 PRINT " A NOTHER ERROR"
20 GOTO 10
```

```
10 PRINT "ANOTHER ERROR"
20 GOTO 10
```

נמקם את הנתב על ה-A במילה ANOTHER. לחיצה על (CTRL) והחזקתו, ולחיצה גם על (INSERT) ארבע פעמים יבצעו את הפעולה. ועכשיו נראה מה קרה: נוספו ארבעה רווחים אחרי הנתב. עכשיו אפשר לתקתק מה שרוצים ברווחים האלה ולשנות ע"י כך את התכנית.
הנה אפשרות אחת:

```
10 PRINT "NOT ANOTHER ERROR"
20 GOTO 10
```

וכעת נדגים את השמוש במקש (DELETE) △

אפשר לסלק את מה שהוספת ע"י הצבת הנתב על ה-N ב NOT ולחיצה על (CTRL) ו- (DELETE) במקביל.

כל לחיצה על (CTRL) ו- (DELETE) תגרום להבלעות אות אחת ושאר השורה יזוז ברווח אחד שמאלה.

לפניך כמה תרגילים בעריכה שיעזרו לך להפעיל בנוחיות את (CTRL) והמקשים הקשורים אליו.

לשם כך צריך לנקות את הזכרון של האטארי, ולתקתק את התכנית הבאה:

```
10 PRINT " I AM TIRED"
20 GOTO 10
```

כעת נשנה את המשפט "I AM TIRED" למשפטים הבאים:

```
" I AM NOT TIRED"
" I AM NOT TRYING"
" I AM TRYING"
" IT IS TRYING"
" IT IS NOT TIRING"
```

ושוב: נקוי הזכרון, ותקתוק:

```
10 PRINT "TA*RI";
20 PRINT "TARI ■ "
30 GOTO 10
```

△ ודאי הבחנת שאפשר להשתמש במקשים (DELETE) (BACKSPACE) גם בלי (CTRL), כאשר רוצים למחוק משהו בתכנית המקורית. לחיצה על מקש זה מוחקת את האות הקודמת. עם החזקת המקש זמן נוסף תמחקנה כמה אותיות.

RUN יריץ את התכנית. פקודת LIST תציג אותה על המסך, ועכשיו אפשר לנסות לערוך אותה בדרכים הבאות.

נשנה את שורה 10:

```
10 PRINT "**++** ■ ";
10 PRINT "**__** ■ ";
10 PRINT "+*+*_+*+* ■ ";
```

נשנה את שורה 20 ונצרף אותה לאחת הצורות של שורה 10:



```
20 PRINT "STARWAR ■ ";
20 PRINT " STAR - WAR ■ ";
20 PRINT " STAR - STARE ■ ";
```

נשנה את שורה 30 בצרוף כמה מהשנויים הקודמים ל:

```
30 GOTO 20
30 PRINT "THE END"
```



קיימת פונקצית עריכה נוספת שמוסיפה ומסלקת שורות במקום רווחים. התכנית הבאה תראה כיצד זה פועל. נמקם את הנתב על הספרה 2 בשורה 20 (באמצעות מקשי העריכה שלמדנו זה עתה):

```
10 PRINT  "*STAR*";
20 PRINT " + STAR+";
30 GOTO 10
```

לחיצה על (SHIFT) ו- (DELETE) ביחד יביאו לכך ששורה 20 תעלם כולה. בצעת עריכת שורות במקום עריכה של אותיות, תוים או רווחים. לחיצה עכשיו על (SHIFT) ו- (INSERT) תיצור מקום לשורה חדשה וניתן יהיה להכניס שורה 20 חדשה, באופן הבא:

(SHIFT) ו- (DELETE) מסלקים שורה
(SHIFT) ו- (INSERT) יוצרים מקום לשורה.

נסלק את שורה 30 בתכנית הבאה:

```
10 PRINT "*+";
20 PRINT "??";
30 PRINT "NOYESNO";
40 GOTO 10
```

ונכניס שורה 30 חדשה שתהיה:




```
30 PRINT "SURPRISE . . I'M BACK!";
```

יש דרך נוספת לזרוק שורה מן התכנית, פשוט לתקתק את מספר השורה שרוצים לסלק ואז ללחוץ (RETURN)

```
30 [RETURN]
```

ועכשיו, הבה נלמד קונץ קטן בעריכה שיחסוך לנו הרבה זמן בחלק מהתכניות שלנו בעתיד.

שימו לב: אנחנו מציגים: "הפעלול הרב-שורתי".
צריך לתקתק NEW, ללחוץ (RETURN),
ואז להכניס את התכנית החד שורתית הבאה:

שלב א':

```
10 PRINT "THIS IS MY ONLY LINE" [RETURN]
```

שלב ב':

תוך כדי שמוש במקשי בקרת הנתב ((CTRL),[↑],[←], וכו'), יש להזיז את הנתב עד שישב מעל הספרה 1 במספר 10. זה צריך להראות כך:

```
[1]0 PRINT "THIS IS MY ONLY LINE"
```

נא לתקתק 2, (RETURN). וכעת שוב, אלא שבמקום 2 נתקתק במקומו 3, וכו'. אחרי שלוש או ארבע פעמים כדאי לתת פקודת LIST. מה קרה? המחשב בצע עותקים רבים של אותה שורה, אבל לכל אחת יש מספר שורה שונה!

התכנית שלך תראה כך:

```
10 PRINT "THIS IS MY ONLY LINE"
20 PRINT "THIS IS MY ONLY LINE"
30 PRINT "THIS IS MY ONLY LINE"
40 PRINT "THIS IS MY ONLY LINE"
```

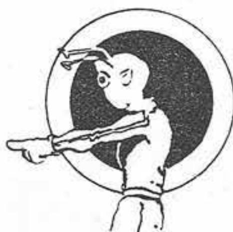
פקודת RUN תריץ את התכנית והיא תדפיס את המשפט הזה עבור כל מספר שורה. בדוגמא שלמעלה קבלנו ארבעה משפטים זהים. בשלב יותר מאוחר של כתיבת תכניות יותר מורכבות, תכונה זו של המחשב תחסוך לך הרבה תקתוק משום שאפשר יהיה לשכפל משפט BASIC בכל מקום בתכנית שלך פשוט ע"י "עריכה" של מספר השורה. ולא לשכוח תמיד ללחוץ על (RETURN) - אחרת לא תכנס השורה לזכרון.

על מנת להכניס שורה חדשה, פשוט ניתן לה מספר בין שני מספרי השורה שביניהם רוצים למקם אותה. כדי להכניס שורה בין 20 ו- 30 בתכנית הבאה:

```
10 PRINT "WW";
20 PRINT "XX"·
30 PRINT "ZZ"
40 GOTO 20
```

פשוט צריך לתקתק

```
25 PRINT "YY";
```





אפשר לתקתק כל מספר שורה בין 20 ו- 30 שעדין לא השתמשת בו.

בודאי הבחנת שמספרי השורות בתכניות שהצגנו עד כה הם כפולות של 10. זה לא היה הכרחי. הם יכלו להיות כפולות של 5, 100, או כל מספר אחר. אפשר היה גם למספר את השורות מ-1 והלאה, אבל כדאי תמיד להשאיר כמה מספרים "פתוחים" בין שורות למקרה שרוצים להכניס שורות מאוחר יותר.

כמה דרכים לתת מספרי שורה בתכנית: 

תכנית מס' 1 - אין מקום לשנויים

```
1 PRINT "X";
2 PRINT "Y";
3 PRINT "Z";
```

תכנית מס' 2 - זה בדרך כלל מספיק

```
10 PRINT "X";
20 PRINT "Y";
30 PRINT "Z";
```

 ודאי הבחנת שתוך כדי קריאה משנה המסך את צבעו מדי פעם. דבר זה נגרם ע"י מנגנון שנבנה אל תוך מחשב ATARI שתפקידו למנוע שריפה של זרחני צבע כתוצאה מהמצאות תבנית אחת על המסך במשך זמן רב מדי.

```
14 PRINT "X";
170 PRINT "Y";
176 PRINT "Z";
```



תכנית מס' 3 -

קצת משונה, אבל יכול לשמש אותנו באותה מידה.

מושגים ☞

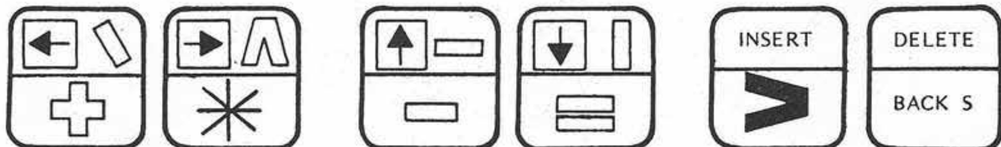
☆ הודעות שגיאה - אלה הן הודעות שנבנו אל תוך ה- BASIC של ATARI או לתוך ההגיון של המחשב שלך. הן מהבהבות על המסך כשבצעת שגיאה נפוצה. כדי לקבל מידע נוסף על שגיאות והודעות שגיאה, רצוי להציץ בנספח א'.



מקשים מיוחדים ☞

☆ (SHIFT) (CLEAR) - לחיצה על (CLEAR) תוך כדי לחיצה על (SHIFT) מנקה את המסך ומציבה את הנתב בפינה השמאלית העליונה, בלי למחוק את הזכרון.

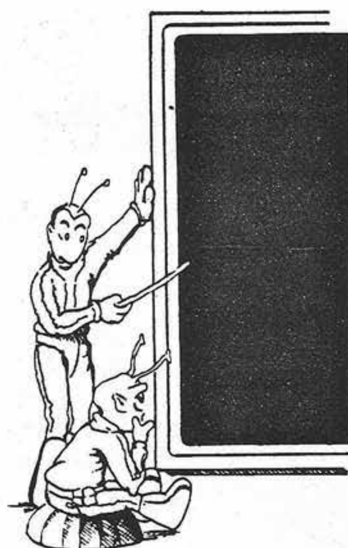
☆ (CTRL) - מקש הבקרה פועל עם המקשים הבאים כדי לתת לך אפשרות לערוך את תכניותיך.



מקשים אלה מזיזים את הנתב. הוא הרבוע הקטן על המסך המציין את המקום שבו ידפיס המחשב את הדבר הבא שתתקתק.

☆ (SHIFT) ו- (INSERT) - מוסיף שורה.
☆ (SHIFT) ו- (DELETE)- מסלק שורה.

☆ 10 (RETURN) - מסלק את שורה 10 מהתכנית. תקתוק מספר שורה ולחיצה על (RETURN) יסלק את השורה הזו מן התכנית.



פונקציית RND(1) **החזקה והזריזה**

אם אי פעם שחקת בקוביות משחק או רולטה או בינגו, הרי שברורים לך העקרונות, או המשמעות של RND (1). אם שלפת פעם שם מתוך הכובע או הוצאת קלף כדי לקבוע מי יהיה ראשון בתור במשחק, גם אז נפגשת ברעיון של RND (1). RND (1) היא פונקציה של מספר אקראי ב-BASIC של אטארי שמשתמשת בהגיון המובנה אל תוך המחשב שלך. פונקציה זו פשוט בוחרת מספר באופן מקרי, או אקראי, כמו בחירה מתוך הכובע, ניעור קופסה של בינגו או שליפה מתוך חפיסת קלפים.

מאחר שהמחשב יכול לשלוט במידע כה רב, ומאחר שהוא מבצע דברים במהירות רבה, הוא יכול גם לבחור מספרים באופן אקראי מתוך מדגם גדול יותר של מספרים משהאדם יכול לבחור באמצעות הכובע או חפיסת קלפים. נתן הוראה למחשב PRINT RND (1).

חשוב לזכור; <u>אין</u> להשתמש במרכאות כאן. מה קבלת?

אנחנו, כשחברנו את הספר, קבלנו 0.2755226953, שבר עשרוני עם 10 ספרות אחרי הנקודה. גם אצלך יתקבל איזשהו מספר עשרוני עם 10 מקומות אחרי הנקודה, בין 0 ו- 1.

ב- BASIC של ATARI מותר לך להשתמש בכל מספר בתוך הסוגרים עבור פונקצית RND: RND (0); RND (1) RND (12345); וכו'. כולם שווים. יחד עם זאת, כדי לשמור על פשטות ועקביות, נשתמש תמיד ב- RND (1).



פקודת RUN לתכנית הבאה תתן לך מטר של מספרים אקראיים:

```
10 PRINT RND(1)
20 GOTO 10
```

זוהי פשוט גירסה חדשה של "תכנית השם" שכתבת קודם.

(BREAK) יעצור את התכנית. בחינה של המספרים תגלה שכולם נמצאים בין 0 ו- 1. המספרים הללו שמושיים מאד למטרות מתמטיות רבות. אבל למטרות משחקים אנו זקוקים למספרים הדומים יותר לאלה שמתקבלים מקוביות או רולטה או קלפים. אנו זקוקים למספרים שלמים. את אלה נקבל בשפת BASIC של אטארי ע"י שמוש בפונקצית INT (מן המלה INTEGER) שמשמעותה מספרים שלמים. אם נשים INT לפני הפונקציה (RND(1 יסולק השבר העשרוני, או יקוצץ ויודפס רק המספר השלם שלפני הנקודה העשרונית. מה לדעתך תעשה התוכנית הזו כשתורץ על ידי RUN?




```
10 PRINT INT(RND(1))
20 GOTO 10
```



משתמשים בסוגרים כמו באלגברה, לציון חלקים של נוסחאות: הסוגריים חייבים תמיד לבוא בזוגות.

פונקצית INT תסלק את הנקודה העשרונית ותדפיס את המספר השלם שלפני הנקודה, שבכל המקרים כאן יהיה 0. מסיבה זו עלינו להוסיף משהו לפונקציה (INT(RND(1 כדי שתהיה שמושית עבורנו במשחקים.



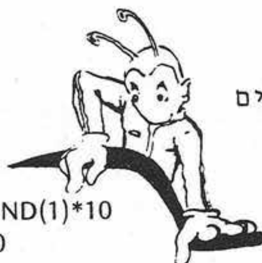
בנקודה זו חשוב להבין מה שאנחנו עושים לפונקציה (1) RND, משום שזו דוגמה לדרך שבה נבנות תכניות מורכבות מחלקים פשוטים. נתחיל עם:

(1) RND שנותן לנו מספר בין 0 ו- 1, עם 10 מקומות אחרי הנקודה.

וכעת אנו מציגים:

מספר חדש שנשתמש בו כדי
לקבל מהמספר המקורי של RND (1)
מספר שהוא גדול מ- 1. נניח
שאנו רוצים להגיע עד 10.ראשית,
אנחנו יכולים **לכפול** את RND (1)
ב- 10. ננסה זאת ונראה אילו מספרים
מייצרת התכנית.

60



```
10 PRINT RND(1)*10
20 GOTO 10
```

פקודת RUN **תריץ** את **התכנית הזו** ואפשר
לראות שמתקבלים כעת מספרים אקראיים
בין 0 ו- 9.99999999.
INT ישנה את התכנית כך שיתקבלו מספרים
שלמים, והתכנית הקצרה תראה כך:

```
10 PRINT INT(RND(1)*10)
20 GOTO 10
```



חשוב להשתמש בסוגרים בזהירות רבה,
כדי למנוע שגיאות תכנות פשוטות. לכל "סוגר"
משמאל חייב להיות בו זוג מימין.רצוי לבדוק כל משפט
המכיל כמה סדרות של סוגריים כדי לראות אם הם
באים בזוגות ולודא שהם אכן נמצאים במקום שרוצים
שיהיו.

תכנית זו תסלק את החלקים העשרוניים של המספרים
ותדפיס רק את המספרים השלמים בין 0 ו- 9. אם נרצה
מספרים שלמים בין 1 ו- 10, נצטרך להוסיף מספר, וכך
תראה התכנית:




```
10 PRINT INT(RND(1)*10)+1
20 GOTO 10
```

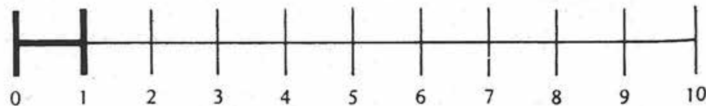


עכשו יש לנו תכנית שמייצרת מספרים בין 0 ו- 10.
אם ברצונך לכתוב תכנית כדי לזרוק קוביה אחת,
עליך לכתוב תכנית RND שתיצר לך מספרים בין **1** ו- 6.
האם יש ביכולתך לכתוב תכנית כזו? כל מה שדרוש הוא
כמה שינויים קלים בתכנית שייצרה מספרים מ- 1 עד 10.

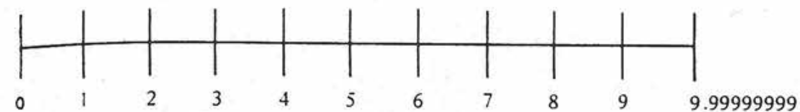
אם היה לך קשה לעקוב אחרי מהלך ההסבר, לא צריך
להכנע. הנה תמונה שמתארת מה קורה.
אם מרגישים שמאבדים את הקשר רצוי תמיד לצייר לעצמך
תמונה של מה עושים עם התכנית. אפשר לתאר
תכנית בצורת דיאגרמה, ושרטוט התכנית הוא מכשיר רב
עצמה בידי המתכנת.

RND (1) נותן המספרים עד למקסימום דיוק של 10
מקומות אחרי הנקודה.

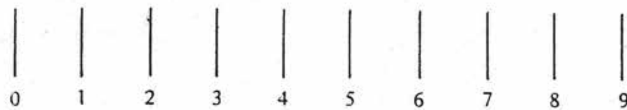




10* RND (1) נותן מספרים בין 0 ו- 9.99999999

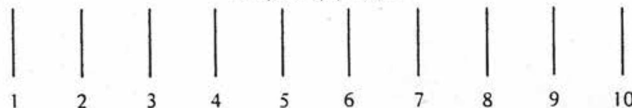


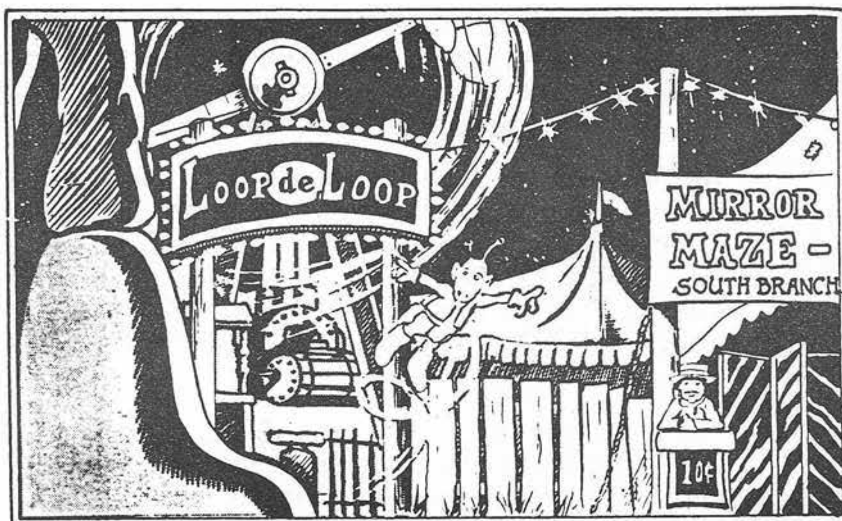
INT (RND (1) * 10) זורק את החלק העשרוני
ומוציא את המספרים השלמים בין 0 ו- 9. המחשב
איננו מוציא את המספר 10 מפני ש 10* RND (1) מגיע
רק עד 9.99999999.





ע"י הוספת 1 למשפט הקודם, נקבל מספרים שלמים
מ- 1 עד 10.





השימוש ב- (1)RND: כמה לולאות והסתעפות, והנה יש משחק.

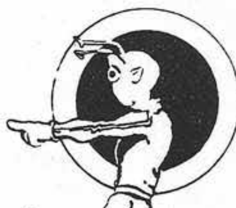


תוספות אחדות לאוצר המלים יאפשרו לך ליצור
כמה משחקי ניחוש מענינים,תוך כדי לימוד ה-BASIC
של אטארי. התוספת הראשונה פשוטה לגמרי:

LET X =

X הוא משתנה;כלומר, ערכו יכול להשתנות בהתאם
לרצונך. נתחיל עם משתנים מספריים ואחר כך נראה
לך כיצד להשתמש במשתני מחרוזת שיכולים להכיל
מלים ואותיות. לפניך דוגמה פשוטה של החלטה לגבי
ערכו של X:

```
10 LET X = 10
20 PRINT X+30
```



מה לדעתך יעשה המחשב כשיקבל פקודת RUN ?
הוא ידפיס 40, משום שהוסיף את ה- X שלך ל- 30.
אם X ישונה למספרים הבאים, מה לדעתך ידפיס המחשב?

X = 90,23,2,34 ,3.007

כדי להתחכם קצת ולהשתמש במה שלמדנו בפרק על RND,
נתן ל X להיות מספר שנבחר באופן אקראי בין 1 ו- 6
(כמו בהטלת קוביה) ע"י כך שנתאר את X באופן הבא:

```
10 LET X=INT(RND(1)*6)+1
20 PRINT X
```

המחשב יבחר באופן אקראי במספר בין 1 ו-6 ואז ידפיס אותו. אם נוסיף את השורה הבאה, ימשיך המחשב לבחור ולהדפיס מספרים אקראיים בין 1 ו- 6 כאילו הטיל קוביות ללא סוף: 30 GOTO 10

עדיף לנסות את התרגילים הבאים לפני שממשיכים הלאה: הבה נתן ל X להיות מספר אקראי בין 1 ו- 20, 1 ו- 4 ו- 1 ו- 52, ואנו מבקשים ממך לכתוב תכנית שתדפיס את המספרים שבחר המחשב למקרה שהוא בוחר רק פעם אחת. ודרישה נוספת של התרגיל היא להוסיף שורה כך שהמחשב נכנס ללולאה אינסופית וממשיך לבחור מספרים באופן אקראי.

הפקודה =LET X יכולה להיות מיגעת: כל פעם שרוצים לשנות את X צריך לכתוב תכנית חדשה, כפי שכתבת קודם לכן. על מנת לפשט את הדברים (והמחשב יוצא מן הכלל בפישוט דברים ברגע שתפסת את העקרון), יש משפט ב-BASIC שמאפשר לך לתת ערך מספרי אחר למשתנה בכל עת שמוצאים לנחוץ. מילת המפתח היא INPUT. פרושה בדיוק מה שהיא אומרת: להכניס משהו. המשהו הזה, כשהוא מספר, מצויין בדרך כלל ע"י שם משתנה כמו X או Y או Z. באופן כזה, INPUT X פירושו שהמחשב ישמור מקום בזכרון שלו עבור X , המספר שתקתקת. הנה התכניונת הבאה:

```
10 INPUT X
20 PRINT X+30
30 GOTO 10
```

התכנית הזו שונה מהתכנית = LET x בכך שהחלפנו את המשפט LET במשפט INPUT. עם הרצת התכנית (ע"י RUN) יופיע הסימן: ? על המסך. עכשו יכולים להכניס כל מספר שהוא לתכנית, נניח 2345, ללחוץ (RETURN) ולקבל את הסכום של המספר הזה, ועוד 30. אחרי שהמחשב מדפיס את הסכום, מופיע הסימן: ? שוב. זה משום שבשורה 30 אמרנו למחשב לחזור ולתת לנו הזדמנות נוספת להכניס מספר חדש למכשיר. בדרך זו נתן לבצע פעולות חיבור בשרשרת (או כל פעולת חשבון מסובבת אחרת), עבור כל מספר של פעולות INPUT בלי לשנות את התכנית. למשל אם רוצים לבנות תכנית שתחלק מספר כלשהו ב 5 ואז תכפיל את התוצאה ב- 12, הנה דרך פשוטה לעשות זאת:




```
10 INPUT X
20 PRINT X/5*12
30 GOTO 10
```



שורה 20, תחשב באופן הבא: קודם חשב X/5; קח את התוצאה והכפל אותה ב- 12. ה - BASIC של אטארי משתמש בכללים המתמטיים המקובלים - כלומר קודם כופל ומחלק. משמאל לימין. ואחכ' מבצע חיבור וחיסור.

כדאי לנסות כמה מספרים ולראות כמה קל לבנות תכנית לחישוב בשרשרת כשמשתמשים במשפט INPUT אחד. אחר כך אפשר להרחיב קצת את העניינים ולראות אם אפשר לפתח כמה תכניות תוך כדי שימוש ב - INT, RND (1), LET X = ו- INPUT. הנה תכנית שמשתמשת בכל המלים האלה של BASIC. מה לדעתך עושה התכנית? פקודת RUN תבדוק אם צדקת :



```
10 LET X=INT(RND(1)*6)+1
20 INPUT Y
30 PRINT X*Y
```

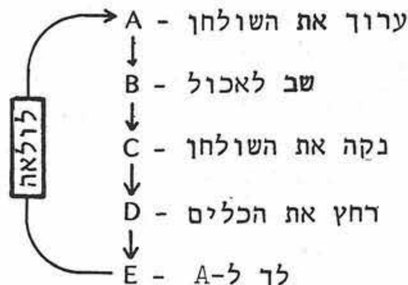
רק עוד מושג אחד, ונוכל לבנות תכנית למשחק ניחוש מספרים. אולי זה נראה עבודה רבה כדי לפתח משחק פשוט, אבל צריך לזכור שעברנו כבר כמעט מחצית הדרך בלימוד המלים הדרושות כדי לשחק, לתכנת ולשנות את רוב המשחקים שבספר. הדוגמה הפשוטה שנביא מיד מציגה את המורכבות במלות BASIC שהצגנו עד עתה.



המלה החדשה ב BASIC (בעצם שתי מלים המתחברות למשפט) היא THEN......IF. המחשב מבין את המושג הזה באופן דומה מאד לצורה שבה משתמשים בו בחיי יומיום. המשפט אומר למחשב שאם (IF) משהו מתרחש אז (THEN) עשה כך וכך, שיכול להיות להסתעף לחלק אחר של התכנית. אם (IF) ה"משהו" הזה לא מתרחש, אז (THEN) המשך כרגיל.

כשענין ההסתעפויות והלולאות יהיה ברור לך לגמרי, תהיה לך היכולת להיות מתכנת בקיא ומיומן. הנה כמה לולאות והסתעפויות מחיי היומיום : איזו טרחה !





E הוא הסתעפות בלתי מותנית וכאן הוא פותח לולאה.

הסבוב היומי



כאן D הוא ההסתעפות הבלתי מותנית שמקיימת את הלולאה.

הסתעפות רטובה:

A - מתחילים בבית

B - לצאת החוצה

C - להחליט

הסתעפות

בהיר / גשם

E - לשחק בכדור

D - מתחילים בבית

ועוד הסתעפות רטובה :

A - לחזור הביתה

B - חוטפים משהו

C - יוצאים החוצה

D - עכשיו מחליטים

הסתעפות

בהיר / גשם

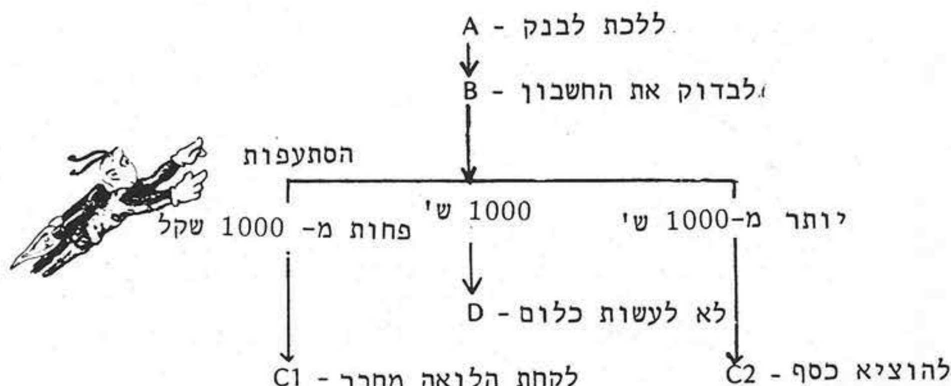
E - לשחק תופסת

D - לך ל-A

לולאה

ההסתעפות הפכה ללולאה מותנית. התנאי הוא, האם יורד גשם - כן, או לא.

שתי הסתעפויות מותנות בלי לולאה



אחרי שראינו כמה דוגמאות של לולאות והסתעפויות, אנו רוצים להציג הסתעפות של IF THEN לתכנית INPUT פשוטה.

נצטרך להשתמש בסימן אחד נוסף: <> שפירושו לא שוה, או שונה מ..... (משמעותו המדוייקת היא "קטן מ.. או גדול מ..!"). יוצרים אותן ע"י תקתוק < ואחר כך > . △

```
10 INPUT X
20 IF X=6 THEN PRINT "THAT IS THE RIGHT NUMBER"
30 IF X < > 6 THEN PRINT "TRY AGAIN"
40 GOTO 10
```

מה לדעתך, אומרת התכנית ? הנסיון לדמיין יעזור לך להבין כיצד בנינו את המשחקים המוצגים בספר וכן יעזור לך לתכנן תכניות באופן עצמאי.

השורה הראשונה, 10 INPUT X, אומרת להכניס מספר, כל מספר שהוא, לתוך הזכרון של המכשיר.
השורה השניה,

```
20 IF X=6 THEN PRINT "THAT IS THE RIGHT NUMBER"
```

אומרת שאם המספר שהכנסת הוא 6, אזי על המכשיר להדפיס "THAT IS THE RIGHT NUMBER".
אם המספר שנבחר (X) איננו 6, המחשב מדלג על שורה 20 ועובר לצעד הבא. זו אחת התכונות היסודיות של המשפט IF THEN וחשוב לזכור אותה.

△ אם כבר מדברים, אז כדאי להציג: < ו- > . משתמשים בהם כמו באלגברה כדי לציין קטן מ (<) וגדול מ (>). => פרושו קטן או שוה ל..., =< פירושו גדול או שוה ל... ו-<> פירושו לא שוה ל.. או שונה מ... הסימנים המתמטיים הם: חיבור + , חיסור - , כפל * , חילוק / והעלאה בחזקה ∧ .

אם התנאי במשפט לא מתקיים, המחשב ממשיך ישר לשורה הבאה בתכנית.

השורה השלישית,
```
30   IF X <> 6
     THEN PRINT "TRY AGAIN"
```
אומרת שאם ה - X איננו 6, המכשיר צריך להדפיס "TRY AGAIN" ולחזור למשפט ה - INPUT כך שהמנחש יכול לנסות שנית.

הנה עוד דוגמא של הסתעפות IF THEN שכדאי לך לנסות לדאות בדמיונך. מה יעשה המחשב עם התכנית הזו? מה דעתך לתקתק את התכנית לתוך המכשיר ולראות אם היא עושה מה שחשבת שתעשה:

```
10 PRINT "WHAT'S MY NUMBER?";
20 INPUT X
30 IF X+2=5 THEN PRINT "YOU GOT IT"
40 IF X+2 < > 5 THEN GOTO 10
```

IF (אם) הצלחת לתאר לעצמך מה עושה התכנית, THEN (אז) יש לך באמת היכולת להשתמש בכל המלים שלמדנו עד כאן כדי לתכנת משחק ניחוש מספרים, ולקשט אותו להנאתך. אבל לפני שאנו מפתחים את המשחק, הבה נחזור על המושגים והמלים שהצגנו בחלק הזה של הפרק.



לומדים!

מושגים ☞

☆ **הסתעפות**- לתכנית יש הסתעפות באותן שורות בהן יכול להתרחש דבר אחד מתוך כמה אפשרויות שונות. ההסתעפויות מיוצגות ע"י משפטי IF THEN.

מילים ב- BASIC ☞

☆ RND(1) - פקודה זו מייצרת מספר אקראי בין 0 ו - 1 (לא כולל 0 ו - 1 עצמם).

☆ INT - פונקציה זו הופכת מספרים עשרוניים למספרים שלמים בדרך הבאה: מכל מספר עשרוני לוקחת הפונקציה רק את המספרים שמשמאל לנקודה. כך, 3 ו- 3.9 ו- 3.999999. יהפכו כולם ל - 3 ע"י הפונקציה INT. אפשר לראות זאת ע"י תקתוק PRINT INT (3) ולחיצה על (RETURN). המחשב ידפיס 3. אם נתקתק PRINT INT (3.999) ואז (RETURN), גם אז ידפיס המחשב 3.

☆ INT (RND (1) * כל מספר) - פקודה זו מייצרת מספרים שלמים אקראיים בין 0 ומספר אחד פחות מן המספר שבחרת (כלומר לא כולל המספר הזה).
על מנת לקבל מספרים אקראיים חיוביים בין 1 - ועד למספר שבחרת עליך לשנות את הפקודה ל-

INT (RND (1) * כל מספר חיובי) + 1

כך, INT (RND (1) *9)

ייצר מספרים אקראיים מ- 0 עד 8 כולל, ו -
INT (RND (1) * 9) + 1

ייצר מספרים אקראיים מ 1 עד 9, כולל.

☆ LET X = פקודה זו נותנת ערך מספרי למשתנה ששמו X.

☆ INPUT X - פקודה זו מרשה למשתמש במחשב לתת ערך מספרי שונה ל - X באמצעות לוח המקשים, בכל פעם שמריצים את התכנית ע"י RUN.

X<Y ;	X קטן מ - Y
X<=Y ;	X קטן או שוה ל - Y
X>Y ;	X גדול מ - Y
X>=Y ;	X גדול או שוה ל Y
X<>Y ;	X שונה מ - Y
X + Y ;	X ועוד Y
X - Y ;	X פחות Y
X * Y ;	X כפול Y
X / Y ;	X לחלק ל - Y
X ^ Y ;	X בחזקת Y

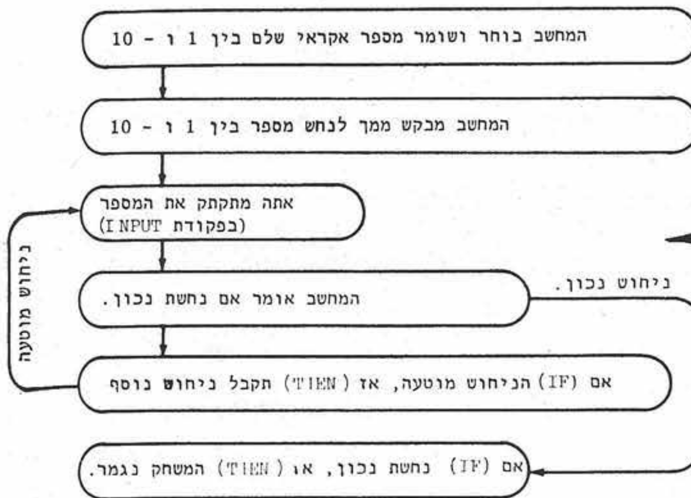
סימני פיסוק

☆ ( ) - משתמשים בסוגריים כמו באלגברה, כדי להפריד בין חלקים בנוסחאות. הסוגרים חייבים תמיד לבוא בזוגות.



סוף סוף משחק: נא לנחש את המספר

אנחנו נתכנת עכשו משחק פשוט של ניחוש מספרים. המחשב יבחר באופן אקראי מספר בין 1 ו - 10, ישמור את המספר בזכרון ואז יבקש ממך לנחש את המספר. נחשת נכון, המחשב יאמר לך שקלעת. אם לא נחשת נכון, יתן לך המחשב לנחש שנית. לפניך המבנה הכללי של התכנית (כדאי לך לאמץ את הרעיון של שרטוט המבנה של התכנית כצעד ראשון שינחה אותך בעבודה):





כדי לכתוב את התכנית הזו בשפה שהמכשיר מבין, נתחיל בכך שהמחשב יבחר במספר בין 1 ו - 10 באופן אקראי. נעשה זאת ע"י הפונקציה (1) RND.

```
10 LET X=INT(RND(1)*10)+1
```

המחשב בחר עכשו מספר ושמר אותו. בצעד הבא צריך המכשיר לבקש ממך לנחש מספר, ואת זה אפשר לעשות ע"י משפט PRINT פשוט:

```
20 PRINT "I HAVE CHOSEN A NUMBER FROM 1 TO
  10. ■■TRY TO GUESS WHAT IT IS" △
```

△ בכל פעם שתתקל בשורה כזו בת יותר מ - 38 אותיות, המשך לתקתק עד סוף השורה הלוגית לפני שתלחץ על מקש (RETURN).

כעת על השחקן לתקתק את הניחוש שלו. עליך להשתמש באות שונה מ - X לציון המשתנה, מכיון ש X כבר שמור עבור המספר שבחר המחשב. נשתמש ב Y, אבל זו יכולה להיות גם כל אות אחרת:

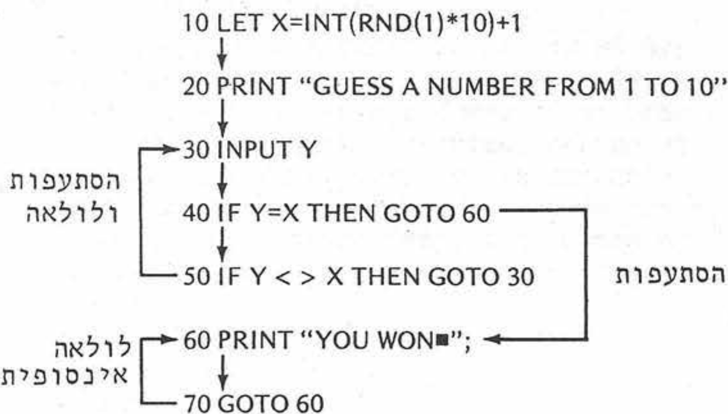
```
30 INPUT Y
```

בשלב הבא עלינו לבנות כמה הסתעפויות של IF THEN, ונהיה מוכנים לשחק:

```
40 IF Y=X THEN PRINT "YOU GOT IT!■GOOD JOB."
50 IF Y < > X THEN GOTO 30
```



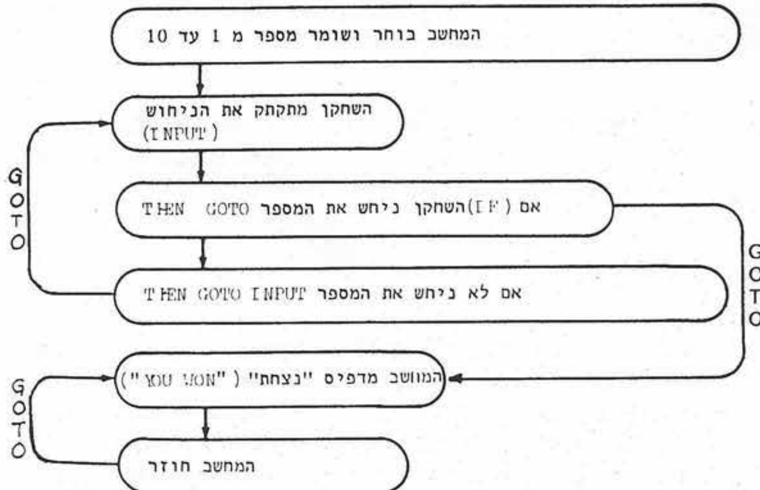
שורה 50 מורה למכשיר לחזור לשורת ה INPUT ולתת לך לנסות שנית. עכשו אפשר להריץ את התכנית ע"י RUN ולשחק. אם יש ברצונך אפשר לשנות את ההודעה במשפט PRINT, להוסיף משהו אחר בסוף המשפט או להעשיר את המשחק בכל דרך שתעלה בדעתך. למשל, אפשר להכניס לולאה אינסופית שתדפיס "נצחת" - "YOU WON", או לבקש שהמחשב יבחר מספר מ 1 עד 25, וכו'. הנה מעין קישוט של המשחק המשתמש בלולאה אינסופית.







וכך פועלת התכנית:



בחלק זה של משחקי מספרים יש כמה גרסאות ו"קישוטים" של המשחק הפשוט הזה. אפשר להסתכל בהם עכשו או לחכות עד שאוצר המלים שלך יתרחב ויהיה באפשרותך לבנות משחק יותר מורכב בכוחות עצמך.

בשלב זה כדאי לנסות לכתוב את התכנית אל תוך המכשיר בלי להסתכל בתכנית שלנו. חשוב לחשוב על השלבים השונים הדרושים למשחק ולנסות לשחזר אותו. אם יתברר שעשית זאת כהלכה, הרי שהגעת לשלב שבו ניתן לעבור למשחקים יותר מורכבים וללמוד כמה מלים חדשות ב-BASIC. אם לא הצלחת, הרי שכדאי לדפדף קדימה בספר, להעתיק כמה מהמשחקים המופיעים בו ולשחק אותם, ואז לחזור לפרק זה בפעם אחרת.

פרק ב'



נושא ווריאציות: מבוא לאמנות העיצוב

המשחקים היפים ביותר, בין אם משחקים אותם במחשב ובין אם לאו, התפתחו ממספר קטן יחסית של נושאים פשוטים. למשל, המשחקים העתיקים "גו" ושח הם משחקים של כיבוש, ומונופול® הוא משחק של שליטה בכסף וברכוש.

מה שמיחד כל אחד מהמשחקים הללו היא הדרך שבה "עוצב" הנושא. כל משחק מציב אותנו בעולם משחקים בעל אופי ואישיות משלו, והשילוב הזה בין פשטות ואלגנטיות הוא שקיים ומתמשך לאורך התקופות ועושה אותנו "מכורים" למשחק. הדבר השובה את הלב במשחקים יפים הוא שאנחנו תמיד ממשיכים למצוא דרכים שונות לשחק בהם ולשנות אותם, ולפעמים עד כדי כך ששינוי החוקים יוצר משחק חדש, השונה לחלוטין מן הקודם.

אנו מקוים שספר זה יעודד אותך לשנות ולהוסיף על משחקים שאנו מציגים כך שהם יהיו משחקים שלך, ושתיצור משחקים חדשים עבור מחשב אטארי שלך שיהיו מקוריים ומרגשים. דרך אחת להכנס לתהליך הזה היא לראות כיצד רעיון מאד פשוט יכול לצמוח ולהתפתח: אפשר לדמות זאת לצפיה בבית ההולך ונבנה.

נושא המשחק שנציג בפרק זה הוא המרוץ הקלסי בין הארנבת והצב. הצב הזקן, כבד התנועה, והמתמיד בדרכו נענה לאתגר של תחרות המוצע לו ע"י הארנבת המהירה. הצב יציב, אך לפעמים אטי עד כדי כאב. הארנבת מהירה אך קופצנית. היא קופצת ממקום למקום, חסרת סבלנות, ובמשל היווני המקורי-גם טפשה. היא רצה מהר מכפי יכלתה ומובסת ע"י העקשנות וההתמדה של הצב. במקביל הממוחשב שלנו של המרוץ, סוף המשחק איננו כל כך צפוי מראש. לכל משתתף יש סכוי והתוצאה משתנה ממרוץ למרוץ. אבל, לפני שנכנס לפרטים בענין הארנבת והצב, עלינו להציג כמה חלקים חדשים של BASIC.

המושג החדש הראשון הוא לולאת FOR-NEXT.
פקודת הלולאה הראשונה שהצגנו היתה פקודת GOTO.
בדוגמה שלנו בנתה הפקודה לולאות כמו זו שלפנינו :

```
10 PRINT "I WISH I COULD DO MORE INTERESTING
   THINGS WITH MY ATARI COMPUTER";
20 GOTO 10
```

® מונופול הוא שם מוגן של חברת PARKER BROTHERS, בע"מ.

ההנחיה GOTO, אם לא נפסיק אותה, תיצור לולאה אינסופית. הפקודות מסוג FOR-NEXT מאפשרות לפקח במדויק על מספר הפעמים שברצוננו לתת למחשב לרוץ דרך הלולאה. למשל, נניח שהיה ברצוננו שהמחשב ידפיס

FOR

PRINT "I'M JUST A MACHINE"

בדיוק חמש פעמים. לפניך שתי דרכים לעשות זאת, באמצעות הפקודות LET, IF-THEN, ו- GOTO :

NEXT

```
10 LET X=0
20 LET X=X+1
30 PRINT "I'M JUST A MACHINE"
40 IF X<5 THEN GOTO 20
```

או:

```
10 LET X=1
20 PRINT "I'M JUST A MACHINE"
30 LET X=X+1
40 IF X<=5 THEN GOTO 20
```

והנה כיצד אפשר לכתוב אותה התכנית ע"י הלולאה FOR-NEXT :

```
10 FOR X=1 TO 5
20 PRINT "I'M JUST A MACHINE"
30 NEXT X
```

אם נבחן את התכנית המשתמשת בלולאה FOR-NEXT, נראה ששורה 10 אומרת למכשיר כמה פעמים לעבור דרך הלולאה. ננסה כמה שנויים. למשל, שנוי שורה 10 ל :

```
10 FOR X=1 TO 10
```

או :

```
10 FOR X=1 TO 1000
```

שורה 20 היא פקודת ה PRINT, שאומרת למחשב מה להדפיס בכל פעם שהוא עובר דרך הלולאה. שורה 30 היא השורה החשובה ביותר : היא סוגרת את הלולאה ודומה למשפט GOTO. הפקודה NEXT X מורה למחשב לחזור להתחלת הלולאה ולבצע אותה שוב עד שיבוצע המספר הנכון של לולאות.

הבה נראה מה קורה בהרצה אם ההוראה NEXT X תוצא מהתכנית :

```
10 FOR X=1 TO 10
20 PRINT "LOOP DE LOOP"
```

המחשב עובר דרך הלולאה פעם אחת ואז עוצר ומחכה לפקודה הבאה. כעת נוסיף :

```
30 NEXT X
```

נריץ את התכנית (RUN) מחדש. מה קרה הפעם ?
בואו נתרגל קצת את הלולאה FOR-NEXT :

תרגיל מס' 1

☆ לכתוב תכנית שתדפיס את שמך בדיוק 12 פעמים

☆ לכתוב תכנית שתדפיס 3 מספרים אקראיים בין 1 ו-6 (מקביל לזריקת קוביה 3 פעמים).

מלת BASIC נוספת שתהיה שמושית לכתיבת תכניות באופן עצמאי ולהבנת תכניות שכתבו אחרים היא REM.



REM באה מן המלה REMark, שפירושה הערה. למעשה היא איננה חלק מהתכנית אלא מהווה הערה עבור המתכנת, כדי לומר לך מה יש בתכנית. קשה להתבונן בתכנית ולנסות לתאר לעצמך מה היא אמורה לעשות, ללא כל רמזים. לכן השמוש במשפט REM הוא בדרך כלל רעיון טוב.
במשחק ניחוש המספרים שלנו אפשר להשתמש במשפטי REM באופן הבא :



```
5 REM NUMBER GUESSING GAME FROM 1 TO 9
10 LET X=INT(RND(1)*9)+1
20 PRINT "I HAVE CHOSEN A NUMBER FROM 1 TO 10.
   TRY TO GUESS WHAT IT IS"
30 INPUT Y
40 IF Y=X THEN PRINT "YOU GOT IT!"
50 IF Y<>X THEN GOTO 30
```



אסור ללחוץ על (RETURN) לפני שמגיעים לסוף "השורה הלוגית". כשנגמר לך המקום על המסך, המחשב שלך ימשיך פשוט לשורה הבאה.

הנה עוד דוגמה למשפט REM :



```
5 REM PRINTING MY NAME MILLIONS OF TIMES
10 PRINT "MY NAME";
20 GOTO 10
```

משפט ה REM איננו משפיע על התכנית בשום צורה שהיא, והמתכנתים יכולים לכתוב את הערות ה REM שלהם בכל סגנון הנוח להם. יש מתכנתים החושבים ש BASIC היא שפה חמורת סבר וחסרת דמיון ולכן הם כותבים הערות REM מצחיקות ויוצאות דופן, ואחרות שהן חמורות כמו השפה עצמה. זו פשוט שאלה של סגנון, של הטבעת החותם האישי שלך על התכנית ועל המכשיר.



אנו זקוקים לעוד 4 פקודות כדי להמשיך עם הצב והארנבת. הפקודות האלה משליכות את מחשב אטארי שלך למצב גרפי ומאפשרות לך לצייר על המסך. פירוש הדבר הוא, שהמסך הופך לניר שרטוט וניתן לשרטט עליו נקודות בדיוק כמו עם ניר ועפרון על ניר שרטוט משובץ. כדי להביא את הטלביזיה שלך למצב שידמה לניר שרטוט גרפי עליך לתת למכשיר הוראת GRAPHICS. פירוט של תשעת המצבים הגרפיים שמחשב אטארי שלך יכול לנצל נמצא בפרק ו'. במשחק שלנו נשתמש במצב גרפי מספר 7, והפקודה

```
10 GRAPHICS 7
```

מאפשרת למחשב לצייר גרף בעל 160 טורים או עמודות, (הממוספרות מ-0 עד 159) ו-80 שורות (ממוספרות מ-0 עד 79). נוסף לכך, יכולות ארבע שורות טקסט להכנס במסך בחלון שמופיע בתחתית מסך הטלביזיה שלך.
מצב גרפי 7 (GRAPHICS 7) יבנה את המסך שלך כפי שמראה הדיאגרמה הבאה:

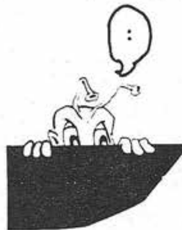


נקודת ה 0,0 של הגרף נמצאת בפינה השמאלית העליונה של המסך. דבר זה שונה מאד מרוב הגרפים שנתקלים בהם, ואשר בהם נקודת 0,0 היא בפינה השמאלית התחתונה.
אבל הפקודה 10 GRAPHICS 7
איננה מספיקה כדי להביא את המחשב לידי כך שיצייר גרף. על מנת שיוכל לצייר דמות צריך המחשב לקבל הוראות לגבי בקרת הצבע ומיקום הנקודה על המסך. צריך לזכור שמחשב אטארי השתלט על מסך הטלביזיה שלך, והמסך יבצע את מה שיורה לו המחשב לעשות. עליך לתת פקודת צבע, COLOR, יחד עם הצבת המסך במצב של GRAPHICS.
תוך זמן קצר נראה לך כיצד אפשר לשחק עם צבע. אבל בינתיים, נניח שהמסך הוא שחור-לבן. כדי לעבור למצב

ולצייר דמות בשחור לבן, זו ההוראה שעליך GRAPHICS 7 לתקתק למחשב:

```
10 GRAPHICS 7
15 COLOR 1
```

שתי השורות האלה אינן תופסות הרבה מקום וה-BASIC של אטארי יכול לצרף שתי שורות BASIC קצרות לשורה אחת באמצעות נקודתים (:). כך אפשר לצרף את השורות 10 ו-15 באופן הבא :



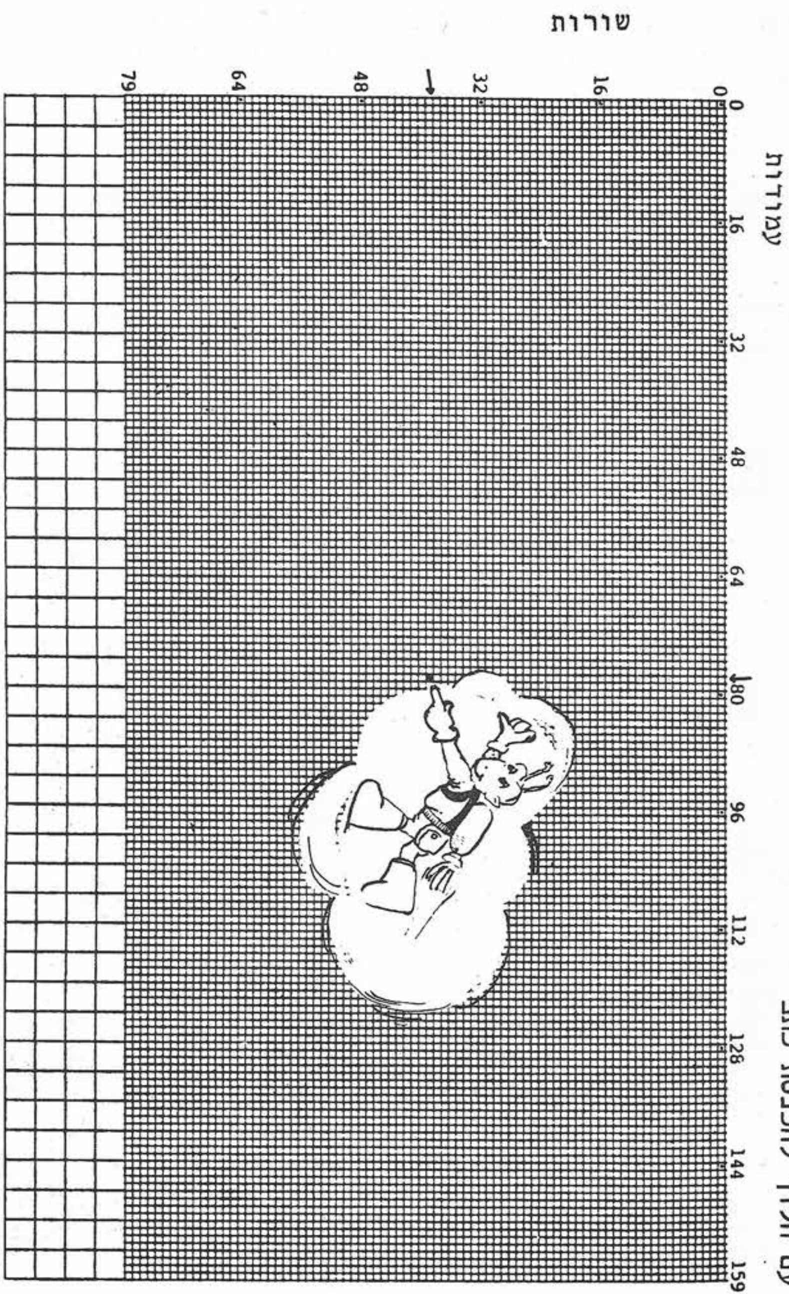

```
10 GRAPHICS 7:COLOR 1
15 [RETURN]
```

מרגע שעשית זאת ניתן לסמן נקודות (PLOT) ולצייר קוים על המסך. סימון הנקודות (PLOT) על המסך נעשה באותו אופן כמו בניר שרטוט גרפי (וצריך כמובן לזכור שהנקודה 0,0 נמצאת בפינה השמאלית העליונה).

מילת BASIC עבור ציור נקודות היא PLOT, כשאחריה שני מספרים. המספר הראשון הוא העמודה שבה נמצאת הנקודה והשני הוא השורה. מפרידים בין שני המספרים ע"י פסיק (,). הנה תכנית פשוטה והגרף שמיצג אותה :



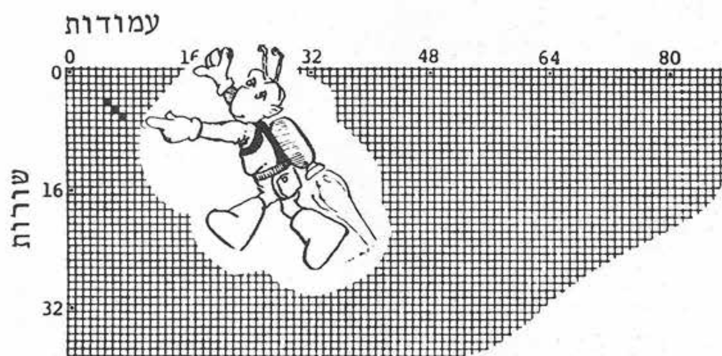

```
10 GRAPHICS 7: COLOR 1
20 PLOT 5,4
30 PLOT 6,5
40 PLOT 7,6
```



מצב גראפי 6 או 7

עם חלון להכנסת כתב

צריך לזכור שהמספר לפני הפסיק הוא מספר העמודה והמספר אחרי הפסיק הוא מספר השורה. חשוב גם לזכור שב- GRAPHICS 7 יש 160 עמודות (0-159) ו-80 שורות (0-79).





אפשר לנסות לדמיין היכן תופענה הנקודות הבאות על המסך ואז להריץ את התכנית כדי לבדוק עד כמה התקרבת בהערכה שלך :

```
10 GRAPHICS 7:COLOR 1
20 PLOT 20,30
30 PLOT 120,10
40 PLOT 10,60
50 PLOT 50,60
60 PLOT 150,10
```

עכשיו אפשר לנסות לצייר כמה נקודות באופן עצמאי ע"י PLOT. תרגיל טוב שיכול לעזור לך להכיר את המסך הוא לכלול PLOT בתכנית שלך ולציין את הנקודה על ניר שרטוט כמו בתמונה, לפני הרצת התכנית(הנספח מראה כיצד להשתמש בניר שרטוט). כעבור זמן מה תתפתח תחושתך לגבי מקום הנקודות על המסך.

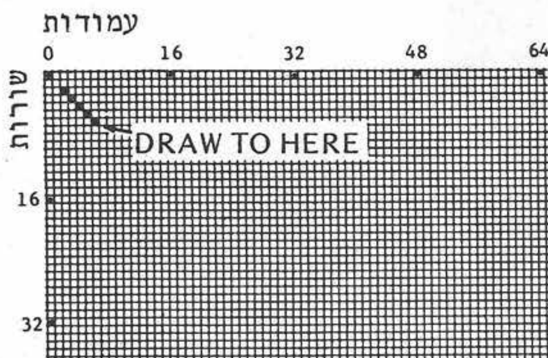


יש עוד פקודה אחת אחרונה שעלינו לקחת בחשבון לפני שנתחיל במירוץ. זוהי DRAWTO. ב-BASIC של אטארי, פקודת DRAWTO מאפשרת לך לצייר קו מנקודה כלשהי על המסך לכל נקודה אחרת על המסך.

לפניך תכנית DRAWTO פשוטה, והגרף שהיא יוצרת :

```
NEW
10 GRAPHICS 7:COLOR 1
20 PLOT 2,2
30 DRAWTO 6,6
```





כדאי לנסות גם את התכניות הבאות.
לפני ההרצה, רצוי להתבונן בהן ולנסות לנחש כיצד
תיראינה על פני המסך :

```
10 GRAPHICS 7:COLOR 1
20 PLOT 150,79
30 DRAWTO 0,0
```



```
10 GRAPHICS 7:COLOR 1
20 PLOT 5,5
30 DRAWTO 10,10
40 PLOT 1,1
50 DRAWTO 6,7
```

איך עושים את זה ?

כאן צריך להשתמש במקש (SYSTEM RESET) או לתקתק
GR.0 אחרי (RETURN).כך אפשר יהיה לקבל הדפס
(LIST) של התכנית המלאה על מסך שלם.ונא לא לשכוח
לתקתק NEW לפני הכנסת כל תכנית!



```
5 REM USING RND(1) WITH GRAPHICS
10 GRAPHICS 7:COLOR 1
20 PLOT 80,40
30 DRAWTO RND(1)*100,RND(1)*50 △
```

```
5 REM ILLUSTRATES HOW REPEATED USE OF
  RND(1) IS INTERESTING AND USEFUL FOR
  GAMES AND GRAPHICS
10 GRAPHICS 7:COLOR 1
20 PLOT 80,40
30 DRAWTO RND(1)*100,RND(1)*50
40 GOTO 20
```

△ האטארי שלך מסמן (PLOT) מספרים אקראיים שלמים
באופן אוטומטי.



לומדים!

אם ההרגשה קצת מבולבלת לגבי השמוש במונחים החדשים הללו, צריך ללמוד את המלים החדשות שהצגנו, ולחזור על הדפים האלה.

מושגים

☆ **מצבים גרפיים-** המחשב הביתי של אטארי יכול להפוך את מסך הטלביזיה שלך למעין ניר שרטוט גרפי שעליו אפשר לצייר. ה-BASIC של אטארי מכיל תשעה מצבים גרפיים. בפרק זה נשתמש ב-GRAPHICS 7 (ראה ברשימת המלים של BASIC למטה). כדי לקבל מידע נוסף עליך לפנות לפרק ו' המטפל בגרפיקה של אטארי.

מילים ב- BASIC

☆ REM - משפט BASIC הבא מן המלה REMark-הערה, תזכורת. מכל מקום, מתעלמת התכנית מן המשפט הזה והוא נמצא בתכנית לשמושך כהערה לגבי מה שהתכנית עושה או אמורה לעשות.

☆ GRAPHICS 7- משפט זה הופך את המסך לניר גרפי בן 160 עמודות (0-159) ו-80 שורות (0-79).

בתחתית המסך ישנו חלון שיכול להציג ארבע שורות של טקסט כתוב.

☆ COLOR 1 - משפט זה אומר למחשב אטארי לבחור בצבע מסוים לציור קוים או סימון נקודות. במסכי טלביזיה שחור-לבן, הוא מספק הנגדה (CONTRAST) טובה בין הדמות לבין הרקע.

☆ FOR-NEXT - זוג משפטים שהולכים תמיד ביחד כדי ליצור לולאה. משפט FOR פותח את הלולאה ומציין כמה פעמים תחזור הלולאה על עצמה. משפט NEXT סוגר את הלולאה כל פעם ומחזיר את התכנית לבצע את הלולאה עד שבוצעה כמספר הפעמים שצויין בראשיתה במשפט FOR. כדי שלולאת
☆ FOR-NEXT תתקיים, חייב להיות משפט NEXT סוגר עבור כל משפט FOR פותח.

הנה דוגמה פשוטה ללולאת FOR-NEXT שחוזרת על עצמה חמש פעמים :

```
10 FOR X=1 TO 5
20 PRINT "HELLO"
30 NEXT X
```

אם תריץ את התכנית הפשוטה הזו, המכשיר ידפיס :

```
HELLO
HELLO
HELLO
HELLO
HELLO
```

☆ PLOT X,Y – משפט זה ממקם נקודה על המסך במקום החיתוך של עמודה X ושורה Y. לתשומת לבך : X ו- Y מופרדים ע"י פסיק. X ו- Y מקבלים ערכים מ-0 דרך כל המספרים החיוביים, כל עוד מספרים אלה נמצאים בתחום מספרי השורות והעמודות האפייניים למצב הגרפי בו נמצא המחשב. עבור GRAPHICS 7, X יכול לקבל ערכים בין 0 ל-159 ו-Y מ-0 עד 79.

(לתשומת לבך : X ו- Y יכולים להיות גם משתנים. למשל, בלולאת FOR-NEXT אפשר לצייר (PLOT) את ארבע הנקודות הבאות : 1,1 2,2 3,3 4,4).

```
10 GRAPHICS 7:COLOR 1
20 FOR X=1 TO 4
30 PLOT X,X
40 NEXT X
```

☆ DRAWTO X,Y – אחרי ציור נקודה ע"י PLOT, אפשר למתוח קו מנקודה זו לכל נקודה מותרת אחרת על המסך באמצעות פקודת DRAWTO.
למשל, התכנית הבאה תצייר קו אלכסוני מן הפינה השמאלית העליונה לפינה הימנית התחתונה של המסך.

```
10 GRAPHICS 7:COLOR 1
20 PLOT 0,0
30 DRAWTO 159,79
```

מקשים מיוחדים ☞

☆ (SYSTEM RESET) - מחזיר את התצוגה על המסך ל-
GRAPHICS 0 (מצב גרפי 0), מנקה את המסך ומחזיר שוליים ומשתנים מיוחדים אחרים לערכי "ברירת המחדל" שלהם.
באופן כללי, אם רצונך רק לנקות את המסך או את חלון הטקסט במצבים הגרפיים, הפעולה המתאימה היא ללחוץ (SHIFT) ו- (CLEAR). ברוב המקרים, לחיצה על (SYSTEM RESET) תשיג את אותה התוצאה, אבל לעתים יש לפעולה זו תופעות לוואי בלתי רצויות.

סימני פיסוק ☞

☆ **נקודותיים**- הנקודותים (:) משמשות לצירוף שתים או יותר פקודות BASIC למשפט אחד השייך לאותו מספר שורה בדוגמה הבאה:

```
10 GRAPHICS 7
20 COLOR 1
```

אפשר לכתוב את שורות 10 ו-20 מחדש:

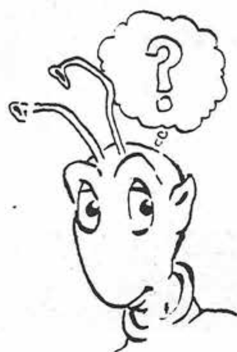
```
10 GRAPHICS 7:COLOR 1
```

צריך לזכור: כל פעם שהשורה הלוגית שמתקתקים ארוכה מ-38 תוים, יעבור מחשב אטארי לשורה הבאה על המסך. (שורות בהדפס של התכנית שמופיעות באורך של שתי שורות או יותר בלי מספר שורה חדש, צריכות להיות מתוקתקות למחשב כאילו היו שורה אחת רצופה).

☆ **פסיק**- הפסיק (,) משמש להפרדה בין המשתנים, או המספרים המציינים את מספרי העמודות והשורות במצבים הגרפיים.



סוף סוף,הארנבת והצב- כמעט!



שתי התכניות הבאות הן אבני בנין למרוץ בין הארנבת והצב. כדאי להתבונן בכל אחת מהן ולנסות לנחש מה יקרה. תכניות אלה יותר מורכבות מאלה שנתקלת בהן עד עתה, אבל לא צריך להבהל אם הן תבלבלנה אותך.צריך לנסות אותן ולשים לב למה שקורה, ושוב לחזור להתבונן בתכניות ולנסות לברר מדוע התרחשו הדברים כפי שהתרחשו. מדוע מיצגת תכנית אחת את הצב והשניה את הארנבת ? הנה כמה רמזים שיעזרו לך. בתכנית הראשונה יש שתי לולאות FOR-NEXT. כיצד הן יכולות לפעול יחד ? צריך לזכור שהלולאה הראשונה (לולאת ה-x) מתחילה, ומיד אחריה מתחילה לולאת Y. נסביר את הדברים בצורה יותר מפורטת בהמשך, אבל כדאי כבר עכשיו לנסות לראות אם הדברים מובנים, ולזכור שמשפטי REM אינם חלק ממה שהתכנית מבצעת. הם כמו הערות המחבר במחזה או בקונצרט, ואינן לביצוע.

```
10 REM * THE TORTOISE FILLS THE SCREEN
15 REM *     STARTING FROM THE LEFT
20 GRAPHICS 7:COLOR 1
30 FOR X=0 TO 159
40 FOR Y=0 TO 79
50 PLOT X,Y
60 NEXT Y
70 NEXT X
```

```
55 REM * AND HERE COMES THE HARE...
60 PLOT (RND(1)*79)+80,RND(1)*79
70 DRAWTO (RND(1)*79)+80,RND(1)*79
80 NEXT Y
90 NEXT X
```

לפני שנמשיך בכמה תרגילים להסבר, יתכן שברצונך לשמור את התכנית באטארי 410-רשם הקסטות שלך. כדי לעשות זאת צריך לטעון את המכשיר בקסטה ריקה, ללחוץ על REWIND כדי לחזור לראשית הסרט ולתקתק במחשב :

CSAVE [RETURN]

המחשב שלך יצפצף פעמיים, כדי לציין שעליך ללחוץ על שני כפתורים ברשם הקסטות. אחרי הצפצופים צריך ללחוץ על [PLAY] ו- [RECORD] ביחד ואז ללחוץ על [RETURN] בלוח המקשים של המחשב.כשתופיע המילה READY על המסך, פירוש הדבר שהתוכנית הוקלטה ונשמרה על הסרט לשימוש חוזר.
ובעתיד, טעינת התכנית מחדש אל תוך המחשב נעשית באופן הבא : מכניסים את הסרט המסוים הזה לרשם הקסטות, לוחצים על REWIND ומתקתקים במחשב: CLOAD [RETURN]



המחשב יצפצף פעם אחת. לחיצה על [PLAY] במכשיר ההקלטה ואז על [RETURN] , מפעילים את התהליך. כשמופיע READY על המסך , מתקתקים LIST כדי לראות את התכנית, ואז מריצים אותה.
כדי לשמור את התכנית על דיסקט (תקליטון), משתמשים בכונן הדיסקטים אטארי 810. רצוי לקרוא את ההוראות במדריך למשתמשים ATARI 810 Operator's Manual
וב- Disk Operating System(DOS) User's Manual
אם הם אינם מצויים בידך, אפשר לפנות לחב' אטארי עפ"י הכתובת :

ATARI Home Computers, P.O. Box 427,
Sunnyvale, CA 94086, U.S.A.

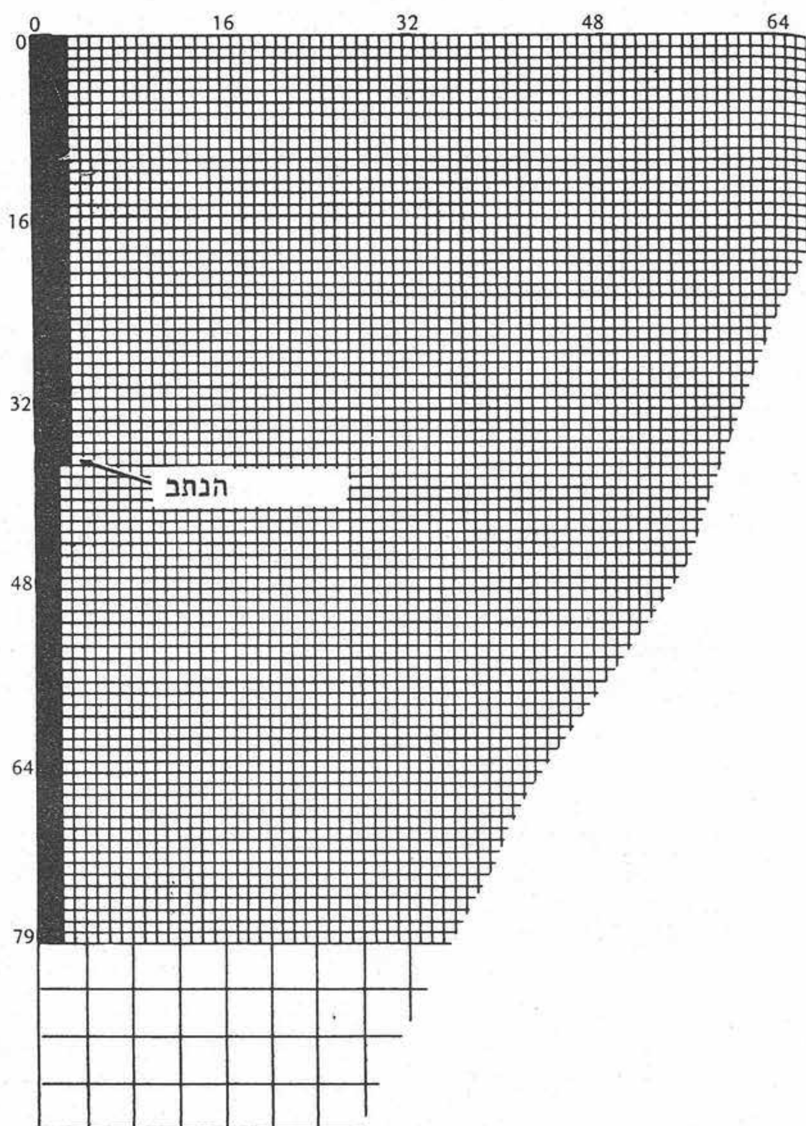
הבה נחזור ונתבונן ביתר תשומת לב ב"הארנבת והצב, גרסה 1".
פרט לשורה 25 יש לנו כמעט אותה התכנית משורה 10 עד 50 כמו שהיתה לנו לפני כן בעמ' 63).
שורה 50 תבוצע 6,400 פעמים (80×80), כשהיא מציירת נקודה אחת בכל פעם. הנקודות מצוירות בסדר קבוע :
0,0; 0,1 ; 0,2 ....0,79 ; 1,0 ; 1,1 ...1,79
וכו'.

שורות 60 ו-70 שונות מאלה שבתכנית הקודמת. הן דומות לשורות 130 ו-140 בתכנית הארנבת בעמ' 67, אבל שינינו את הערכים של X: RND(1)x79)+80) ייצר ערך אקראי בין 80 ו-159. לכן ערך זה של x תמיד יציב נקודה בחלק הימני של המסך. זה מה שגורם לתכניתנו לתת את התחושה של "מירוץ", משום שאנו צופים בצב האיטי, המתמיד, (אשר בעצם נע מהר מאד לגבי צב!) כשהוא ממלא לאיטו את המחצית השמאלית של המסך בעוד הארנבת קופצת ממקום למקום ובחוסר עקביות מציירת קוים וממלאת את הצד הימני של המסך. במבט ראשון כאילו ברור איזה צד יתמלא תחילה. אבל יתרונה ההתחלתי של הארנבת נסחף בהדרגה על ידי היסודיות של הצב, אשר אינו מחמיץ אף נקודה !

נתבונן בשורה 25. כשנותנים פקודת PRINT תוך כדי מצב גרפי, מודפס הטקסט ב"חלון" הקטן בתחתית המסך. זהו מקום מצוין להדפיס הודעות וגם להכניס ערכים באמצעות INPUT או להדפיס (PRINT) מידע נחוץ לתכנית. אפשר להכניס כל הודעה שהיא לפקודת PRINT בשורה 25 או להשמיט אותה לגמרי.

שימו לב, שיש לנו שתי לולאות FOR-NEXT בתכנית.
יש כאן לולאה אחת נמצאת כולה בתוך הלולאה השניה.
למצב זה קוראים "לולאות מכוננות". הלולאה שבה משתנה ערכו של Y מבוצעת שוב ושוב, עד סופה - עבור כל ערך של X. זו גם הסיבה שהמשפט NEXT Y חייב לבוא לפני NEXT X.

על מנת להבין את המשמעות של לולאות מכוננות (לולאה בתוך לולאה) הבה נפשט את הענין ונניח שהמסך מכיל רק 17 עמודות ו-17 שורות (שניהם ממוספרים מ-0 עד 16).
נראה כעת מה תעשה התכנית של לולאות מכוננות :



תכנית זו תמלא את כל המסך : היא תתחיל בפינה השמאלית העליונה ותרד לאורך עמודה מס' 0 עד שתגיע לתחתית. אז תתחיל שוב למעלה, ותרד דרך עמודה 1. התמונה מראה כיצד יראה המסך באמצע העמודה השלישית. תכנית זו מתנהלת לאיטה, נקודה אחר נקודה באותה עקביות כמו הצב.

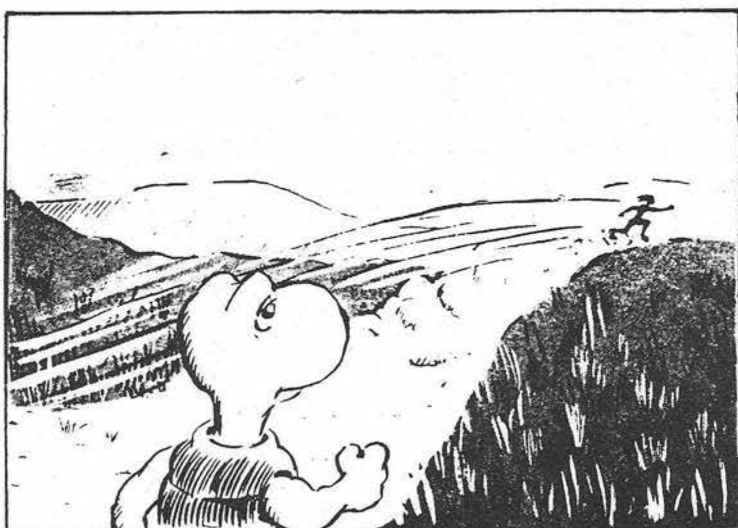
```
100 REM * THE HARE FILLS THE SCREEN
105 REM *    USING RANDOM POINTS
110 GRAPHICS 7:COLOR 1
120 PLOT RND(1)*159,RND(1)*79
130 DRAWTO RND(1)*159,RND(1)*79
140 GOTO 120
```



תכנית זו מציירת (PLOT) נקודה אקראית על המסך, בוחרת נקודה אקראית אחרת ואז מציירת קו מן הנקודה הראשונה לשניה. ע"י שימוש בלולאת GOTO בשורה 140, התכנית חוזרת על כל התהליך מחדש, וממלאת את המסך עם קוים אקראים שקופצים הנה ושמה בדומה לארנבת.

האופן שבו בנינו את התכניות האלה ונתנו להן מספרי שורות מאפשרים לך להריץ אותן בזו אחר זו בלי לתקתק NEW שימחק אותן מן הזכרון. אם נתקתק משורה 10 עד 70, נראה קודם את הצב מתאמן לקראת המרוץ. ואז, הרצת התכנית משורה 100 עד 140 תראה כיצד הארנבת מתקדמת בקפיצות.△

△ על מנת להפסיק כל אחת מן התכניות האלה באמצע, צריך ללחוץ על (BREAK). כדי להמשיך, צריך לתקתק CONT, ואז (RETURN). התכנית הראשונה תפסיק לבסוף בעצמה, אך השניה תמשיך ללא סוף גם אחרי שהמסך יתמלא, משום שאין בה כל משפט המורה לה מתי להפסיק.



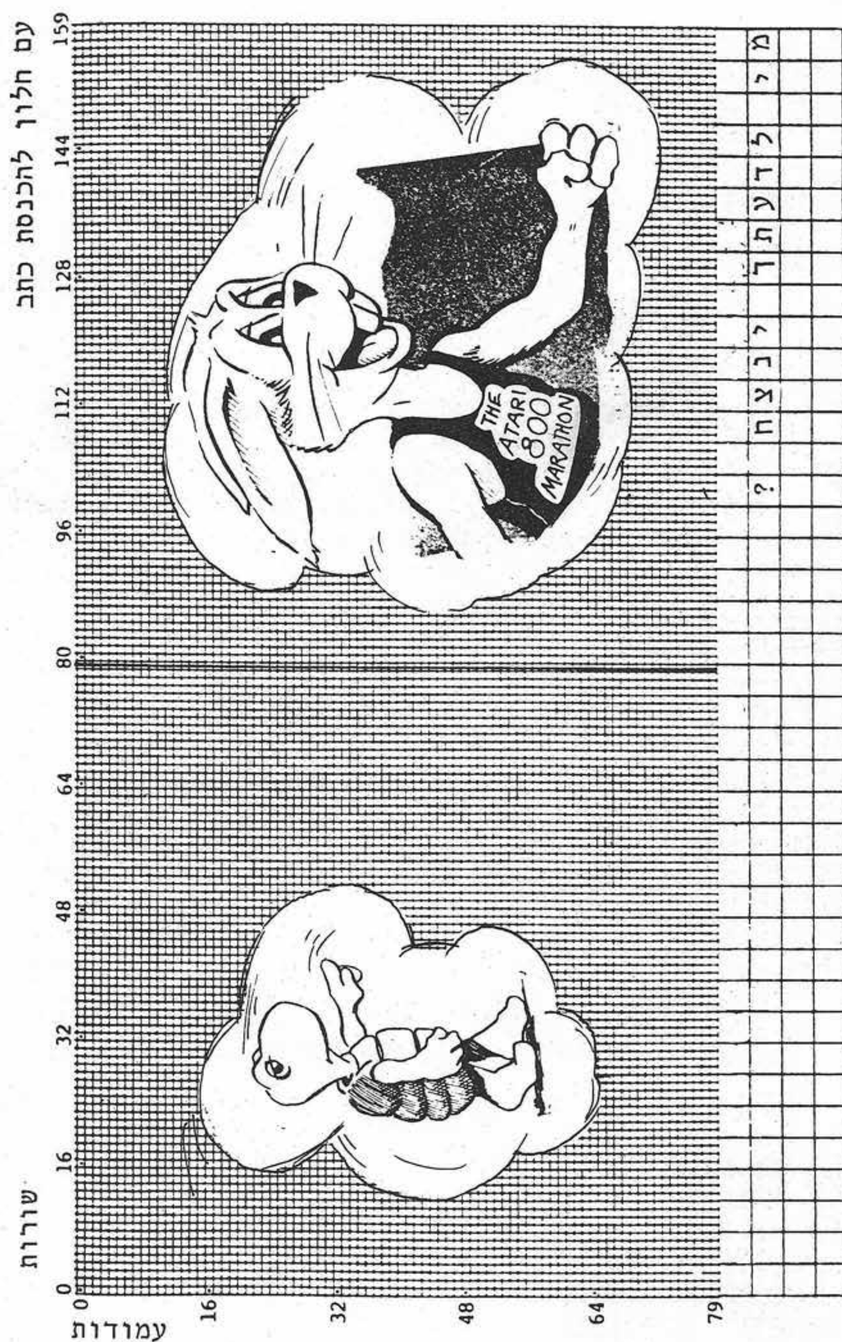
המרוץ: גרסאות 2,1 ו-3

הבה נגש למירוץ, נקח את שתי אבני הבנין ונשלב אותן בתכנית אחת שבה ישתמש הצב בחצי השמאלי של המסך והארנבת תשתמש בחצי הימני. לכן, כל הנקודות על המסך עד עמודה 79 מגדירות את התחום של הצב וכל אלה מעמודה 80 עד 159 מגדירות את התחום של הארנבת. נוסף לכך, נצטרך לשלב את התנועות של הצב והארנבת כך שירוצו בעת ובעונה אחת. הדרך לעשות זאת היא להכיל את מהלך התנועה של הארנבת בתוך הלולאה שמגדירה את מהלך התנועה של הצב ; כך יבצע הצב תנועה אחת בתורו , אח"כ הארנבת ואז תתחיל הלולאה מחדש.

לפניך תכנית שבונה מירוץ בין הצב והארנבת. זו התכנית הקשה ביותר שנתקלת בה עד כה, לכן צריך סבלנות. שינינו את המסך כך שכל בעל חיים נע רק על מחצית מפני השטח, ושלבנו את תוריהם בתוך אותה לולאה.

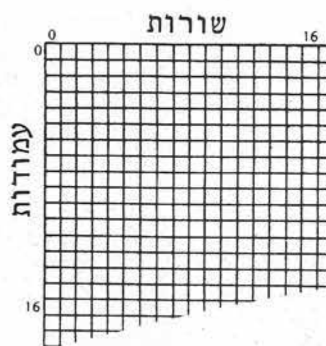
```
10 REM *  THE TORTOISE & THE HARE
15 REM *          VERSION 1
20 GRAPHICS 7:COLOR 1
25 PRINT "WHO DO YOU THINK WILL FINISH
 FIRST?"
30 FOR X=0 TO 79
40 FOR Y=0 TO 79
50 PLOT X,Y:REM * THIS IS THE TORTOISE
```

Graphics Mode 6 or 7

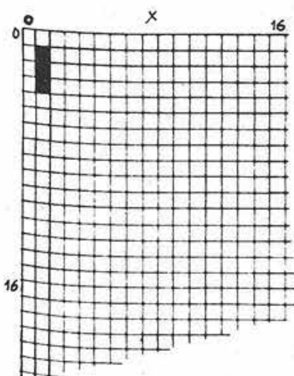


הארנבת והצב גרסה מס' 1

```
10 GRAPHICS 7:COLOR 1: REM SIMPLIFIED
20 FOR X=1 TO 5
30 FOR Y=1 TO 3
40 PLOT X,Y
50 NEXT Y
60 NEXT X
```

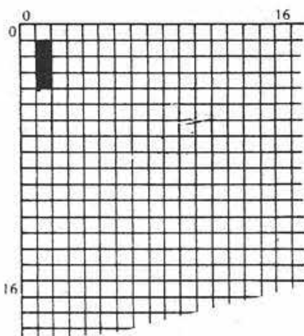


מה שתכנית זאת אומרת הוא לקחת את הערך הראשון של X ואז ללכת ללולאת Y ולבצע את הפקודות בלולאת Y. הלולאה של Y חוזרת על עצמה שלש פעמים. כך, עבור הערך הראשון של X (X=1), ההוראה היא לעבור דרך הלולאה של Y כפי שהדבר מתואר בטבלה ובדיאגרמת המסך :



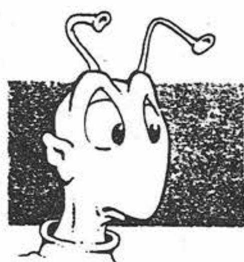
לולאת Y אומרת: PLOT X,Y	עמודות Y	שורות X
1,1	1	1
1,2	2	
1,3	3	

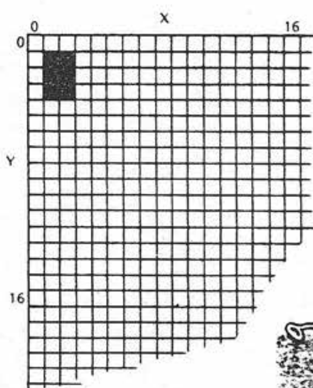
עכשיו יצאנו מלולאת ה- Y ועלינו ללכת ל- X הבא. ה- X הבא לוקח אותנו שוב לתוך לולאת Y עם ערך חדש ל- X.



לולאת Y אומרת: PLOT X,Y	ROW Y	COLUMN X
1,1	1	1
1,2	2	
1,3	3	
2,1	1	2
2,2	2	
2,3	3	

עליך להשלים את הטבלה הבאה ולמלא את המסך כדי לראות כיצד יראה כאשר לא יהיו יותר ערכים עבור X והתכנית תעצור





שורות X	עמודות Y	לולאת Y אומרת: PLOT X,Y
1	1	1,1
	2	1,2
	3	1,3
2	1	2,1
	2	2,2
	3	2,3
3	1	
4		
5		



לפני שנמשיך הלאה, כדאי להריץ את המרוץ כמה פעמים יחד עם חברים. האם הצב מנצח תמיד ? כיצד נעשה את המשחק מעניין יותר ? התשובה לשאלה הראשונה היא "לא", אבל הסיכויים הם כל כך גדולים לרעת הארנבת, שנצטרך לחכות די הרבה זמן עד שנראה אותה מנצחת. יחד עם זאת, ביכולתנו לשנות את התכנית כך שהמשחק יראה יותר הוגן ע"י שנאיץ את תנועותיו של הצב. אפשר להשתמש בתכנית זו כמשחק התערבות ידידותי עם החברים. במבט ראשון, רוב האנשים יהמרו מיד על הארנבת !
כל מה שנעשה הוא לשנות את השורות 40 ו-50 בגירסה 1 של הארנבת והצב. במקום שהצב שלנו יצייר כל נקודה, נגרום לו לצייר קטעי קוים - כמו הארנבת, אבל בצורה מסודרת.





```
40 FOR Y=0 TO 39
50 PLOT X,Y:DRAWTO X, Y+40
```

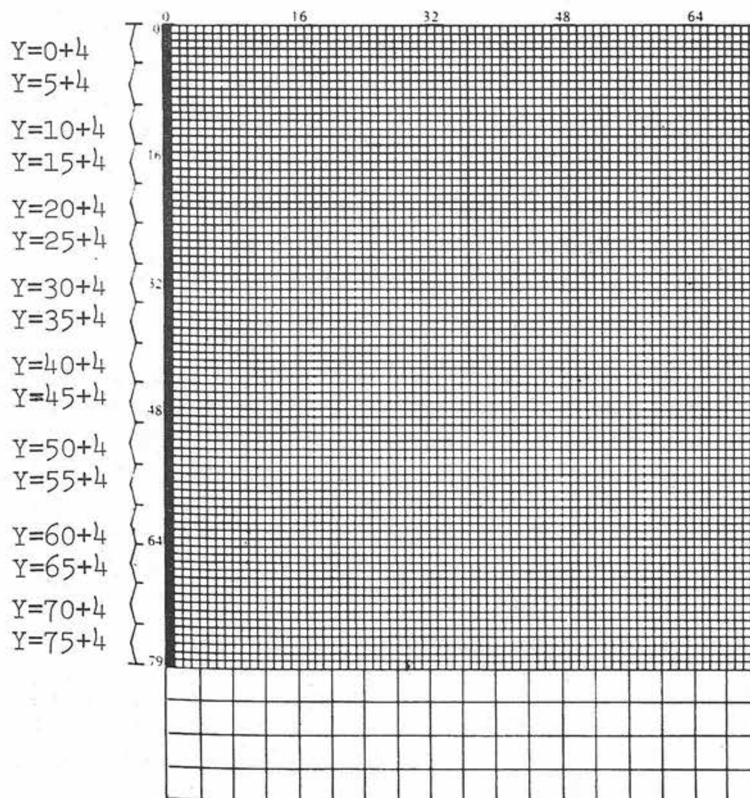
ההבדל נראה בהרצת הגירסה החדשה הזו. הצב עדיין איטי למדי, אבל כמעט תמיד יצא מנצח.

כדי להאיץ את הצב, עלינו להציג חלק חדש של מושג הלולאה FOR-NEXT. זוהי המלה STEP.נשנה את השורות 40 ו-50 באופן הבא :

```
40 FOR Y=0 TO 75 STEP 5
50 PLOT X,Y:DRAWTO X,Y+4
```

מה ש- STEP עושה הוא לשנות את המידה שבה גדל Y כל פעם בלולאה. במקום להוסיף 1 ל-Y כל פעם שהוא מגיע ל- NEXT Y , מוסיפה התכנית החדשה הזו 5 ל-Y, כך יהיו הערכים של Y: 25, 20, 15, 10, 5, 0 וכו' עד 75.





הצב יצייר כעת קוים קצרים שאינם עולים זה על זה, וע"י כך יחסוך זמן רב. זה עדיין יראה כנצחון ברור של הארנבת, וההימור ישאר כנראה כמו קודם. אבל אם נמשיך עוד קצת לשחק עם שורות 40 ו-50, נוכל לנסות עד שהמרוץ החזותי יראה בערך בעל סכויים שוים. כדי להתחיל, נשנה כך :



```
40 FOR Y=0 TO 70 STEP 10
50 PLOT X,Y:DRAWTO X,Y+9
```

אפשר גם לשפר את הצורה החזותית של המשחק ע"י הודעות שתודפסנה על המסך, עבור ערכים מסוימים של X.הדפסי התכניות בעמוד הבא מראים איך "לקשט" קצת את הענין. אנו משתמשים במה שאפשר לכנות לולאה FOR-NEXT "ריקה". לולאה ריקה גורמת להפסקה או דחיה בתכנית ואפשר לקרוא לה "לולאת הפסקה" או "פסק זמן". הנה דוגמה :



```
140 FOR WAIT=1 TO 800 : NEXT WAIT
```

נוסיף את השורה הזו לתכנית ואז פקודת RUN . שמת לב שיש דחיה לפני תחילת המרוץ ? אפשר להשתמש בלולאת הפסקה כדי לשמור הודעה או דמות על המסך למשך כמה זמן או כדי להאיט את פעילות התכנית. הנה כמה דוגמאות שכדאי לך לנסות בשלב יותר מאוחר, כשנגמור עם המרוץ. למשל, מה תעשה התכנית הזו, שהאיטיות שלה ממש מרגיזה?

```
10 PRINT "HELLO";
20 FOR WAIT=1 TO 1500 : NEXT WAIT
30 GOTO 10
```

ניתן להגביר את המהירות של אותה התכנית ע"י קיצור ההפסקה - במקרה זה, ע"י הורדת 1500 ל-50 :

```
10 PRINT "HELLO";
20 FOR WAIT=1 TO 50 : NEXT WAIT
30 GOTO 10
```

בחזרה אל הארנבת והצב. כדאי לשים לב שההדפס הבא של התכנית נראה שונה. למעשה זוהי הרחבה של גירסה 1, עם כמה תוספות ל"עיצוב".

הדפס זה של התכנית שוחזר במיוחד כך שאפשר יהיה לקרוא אותה בקלות. שיטה זו של שיחזור תכניות תשמש אותנו מכאן ואילך. כמה שורות בהדפסה תהיינה ארוכות יותר ממספר התוים שעל המסך. במקרה כזה, המחשב יפצל את השורה ובאופן אוטומטי ימשיך אותה בשורה הבאה על המסך. המקשים המיוחדים יבואו בסוגריים, כך שחשוב לזכור ללחוץ על (RETURN) בסוף כל שורה לוגית של התכנית. לא לשכוח גם, שכל שורה בתכנית מתחילה במספר שורה.

WAIT הוא פשוט שם ארוך למשתנה, כמו X או Y או A.

לבסוף, הכנסת משפטי REM היא על פי שיקול דעתך. כללנו אותם כדי לעזור לך להבין חלקים שונים בתכנית,אבל הם לא הכרחיים.

כשמופיע (CLEAR) בתכנית, פירושו :
ללחוץ על מקש (ESC),
ללחוץ ולהחזיק את (SHIFT) ובאותו זמן
ללחוץ על (CLEAR).

```
10 REM *       TORTOISE & THE HARE
20 REM *             VERSION 2
30 REM *      COPYRIGHT (C) 1980 BY
40 REM * TED M. KAHN & HERBERT R. KOHL
50 REM *
100 GRAPHICS 7:COLOR 1
120 PRINT "[CLEAR]A RACE: THE TORTOISE
 VS. THE HARE"
140 FOR WAIT=1 TO 800:NEXT WAIT
145 REM * THE ABOVE LOOP DOES NOTHING
  BUT DELAY BETWEEN PRINT STATEMENTS
180 PRINT "[CLEAR]ON YOUR MARKS,  GET
SET,  GO!!!"
190 FOR WAIT=1 TO 500:NEXT WAIT
200 REM * THE RACE
210 FOR X=0 TO 79
215 REM * THE TORTOISE
220 FOR Y=0 TO 75 STEP 5
250 PLOT X,Y:DRAWTO X,Y+4
260 IF X=10 THEN PRINT "[CLEAR] PLACE
YOUR BETS, FOLKS!"
262 IF X=40 THEN PRINT "[CLEAR]WANT TO
 CHANGE YOUR BETS?..."
264 IF X=60 THEN PRINT "[CLEAR]  IT'S
DOWN TO THE WIRE, FOLK..."
```

```
266 IF X=75 THEN PRINT "[CLEAR]AND THE
 WINNER IS...???"
268 IF X=79 THEN PRINT "[CLEAR] GUESS
WHO!!!"
300 REM * THE HARE
310 PLOT (RND(1)*79)+80,RND(1)*79
320 DRAWTO (RND(1)*79)+80,RND(1)*79
330 NEXT Y
340 NEXT X
400 PRINT "[CLEAR]  ONCE AGAIN, ORDER
TRIUMPHS OVER CHAOS!"
500 FOR WAIT=1 TO 1000:NEXT WAIT
```



השורות 120, 180, 260, 262, 266,264, 268 ו-400 מכילות כולן פקודת PRINT להודעות שיופיעו כהערות בחלון הטקסט כשהתכנית תורץ.כאשר (CLEAR) משובץ במשפט PRINT עליך ללחוץ על (ESC) ואז ללחוץ ולהחזיק את (CTRL) ובאותו זמן ללחוץ על (CLEAR). זה ינקה את החלון בתחתית המסך לפני הופעת ההערה הבאה.

כעת, לפני שנמשיך לתכנית הבאה, ודאי יהיה ברצונך לשמור את התכנית האחרונה על רשם הקסטות או על דיסקט.

יש גמול לסבלנותך ולהתמדה שלך ! לפניך גירסה 3 של המירוץ בין הארנבת והצב. יתכן שהתכנית לא תהיה מובנת מיד בהתחלה, אבל היא תראה לך כמה מן הדברים שניתן לעשות עם מחשב אטארי.
גירסה זו משתמשת בשני צבעים ויש לה "דף כותרת".
היא גם מראה לך כמה מהדברים שאפשר יהיה לעשות אחרי קריאת הפרקים הבאים בספר.
כדאי להעתיק את התכנית, להריץ אותה ולהוסיף אותה לאוסף שלך.

כל מחשב אטארי יציג 24 שורות של טקסט על המסך בכל פעם. כאשר יש יותר מ-24 שורות, מוזז הטקסט למעלה מחוץ למסך. כדי להפסיק את תזוזת (scrolling) התכנית החוצה מן הקצה העליון של המסך, צריך ללחוץ ולהחזיק את מקש (CTRL) ובאותו זמן לתקתק את המספר 1.
כדי להמשיך את הדפסת התכנית (LISTing)צריך פשוט ללחוץ על (CTRL) ולתקתק שוב 1. נוסף לכך, הדפסי התכניות בספר זה מתאימים לשוליים קבועים במחשב אטארי שלך, כלומר-יש 38 תוים לשורה במסך, שורה מודפסת אחת של תכנית בספר מקבילה לשורה אחת על המסך.



```
10 REM *        TORTOISE & THE HARE
20 REM *              VERSION 3
30 REM *      COPYRIGHT (C) 1980 BY
40 REM * TED M. KAHN & HERBERT R. KOHL
50 REM *
100 GOSUB 1000:REM * ADD A TITLE PAGE
110 GOSUB 1100:REM * ADD SUSPENSE
120 GRAPHICS 7
125 POKE 752,1:REM * TURN CURSOR OFF
130 SETCOLOR 0,3,4:SETCOLOR 2,15,4
135 PRINT "WHO WILL FINISH FIRST?"
```



```
200 REM *
202 REM * THE RACE BEGINS...
204 REM *
210 FOR X=0 TO 79
220 FOR Y=0 TO 80 STEP 4
230 REM *
232 REM * THIS IS THE TORTOISE
234 REM *
240 PLOT X,Y:DRAWTO X,Y+3
250 REM *
252 REM * AND HERE IS THE HARE
254 REM *
270 COLOR 3
280 PLOT (RND(1)*79)+80,RND(1)*80
290 DRAWTO (RND(1)*79)+80,RND(1)*80
300 IF X=20 THEN GOSUB 1200
310 IF X=40 THEN GOSUB 1400
320 IF X=60 THEN GOSUB 1600
350 NEXT Y
360 NEXT X
400 GOSUB 2000:REM * WINNING MESSAGE
402 REM * NOTE THAT THIS PROGRAM
404 REM * ASSUMES THE TORTOISE WILL
406 REM * ALWAYS WIN!  WILL IT?!
408 REM *
410 GOSUB 5000:REM * PAUSE
990 PRINT "THE END"
995 POKE 752,0:REM * TURN CURSOR ON
999 END :REM * END OF MAIN BODY OF
  PROGRAM AND BEGINNING OF SUBROUTINES
```



```
1000 REM * ADD A TITLE PAGE
1010 GRAPHICS 1
1015 POKE 752,1:REM (CURSOR OFF)
1020 POSITION 4,3
1022 PRINT #6;"THE TORTOISE"
1024 POSITION 8,5:PRINT #6;"AND"
1026 POSITION 6,7:PRINT #6;"THE HARE"
1099 RETURN :REM BACK TO MAIN PROGRAM
1100 REM * INITIAL SUSPENSE MESSAGE
1110 PRINT "[CLEAR]ON YOUR MARK..."
1115 GOSUB 5000
1120 PRINT "[CLEAR]GET SET..."
1125 GOSUB 5000
1130 GRAPHICS 2+16:REM (BLACK-OUT)
1135 POSITION 8,6:PRINT #6;"GO!!"
1140 GOSUB 5000
1199 RETURN :REM RETURN TO PROGRAM
1200 REM * INTERMITTENT MESSAGES
1210 PRINT "[CLEAR]        ARE YOU STILL
 SURE?"
1299 RETURN
1400 REM * MORE INTERMITTENT MESSAGES
1410 PRINT "[CLEAR]DO YOU WANT TO CHAN
GE YOUR BET?"
1499 RETURN
1600 REM * ANOTHER SUSPENSE MESSAGE
1610 PRINT "[CLEAR]WHAT A THRILLER!
 WHAT SUSPENSE!"
1699 RETURN
2000 REM * THE WINNER
2010 PRINT "[CLEAR]  THE TORTOISE WINS
 AGAIN!"
2099 RETURN
5000 REM * DELAY (PAUSE) ROUTINE
5010 FOR WAIT=1 TO 600:NEXT WAIT
5099 RETURN
```





ודאי הבחנת שכמה מהמלים ב-BASIC ורעיונות בתכנות אינם מוכרים לך. הם מתקדמים יותר, והיקף ההסברים בספר איננו כולל את כולם.

את מילות BASIC החדשות GOSUB ו-RETURN נסביר בפרק הבא, אבל ניתן כעת הסבר מקוצר, כדי שנבין על מה מדובר :



משפט GOSUB אומר למחשב ללכת למספר השורה שמצויין במשפט, ממש כמו משפט GOTO.
כשהתכנית מגיעה למשפט RETURN (לא לבלבל אותו עם המקש (RETURN) המתאים לאותו GOSUB , היא חוזרת למקום שממנו יצאה בחלק הראשי של התכנית. באופן כזה, אין צורך לעקוב במיוחד היכן עזבת את התכנית הראשית ולהיכן לחזור. צורה זו של כתיבת תכניות נקראת "תכנות מודולארי", ומקובלת בדרך כלל כסגנון טוב של תכנות.



למשל, אחרי הרצת התכנית הזו ודאי יהיה ברור יותר כיצד אפשר להפוך משחקים פשוטים למורכבים יותר, וכיצד ניתן לעצב ולעטר אותם כך שימשכו יותר את העין. נסיים פרק זה בכמה "השלמות לעיצוב" שניתן להוסיף למשחקים. רבות מהן כוללות אוצר מלים ב- BASIC ומושגים בתכנות שכבר הצגנו. בשלב זה, צריך כבר שיהיה ביכלתך ליצור את הגירסה האישית שלך למרכיבי העיצוב האלה, וכן להשתמש באלה היותר מורכבים ויתכן שאף לא לגמרי מובנים לך כרגע.

מושגים ☞

☆ **חלון טקסט** - כשנותנים הוראת PRINT תוך כדי מצב גרפי, מופיעה ההדפסה ב"חלון טקסט" קטן בתחתית המסך. זהו מקום אידיאלי להודעות והערות, וכן למתן ערכים או מידע אחר הדרוש ע"י התכנית.



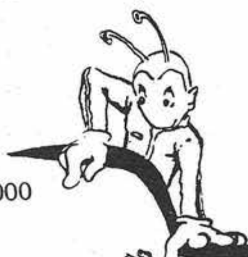
☆ **לולאות מכוננות** - האפשרות להשתמש ביותר מלולאה אחת, לפעמים גם לולאות בתוך לולאות אחרות, היא אחת התכונות רבות העצמה של שפות התכנות במחשב. הדבר חוסך תכנות נוסף ומקל על משימות רבות, ביחוד אלה הדורשות ביצוע פעולה מסוימת שוב ושוב עבור ערכים שונים. יחד עם זאת, לולאות מכוננות יכולות לבלבל את המתכנת המתחיל. להלן דוגמה של לולאה מכוננת :



```
10 REM NESTED LOOP
20 GRAPHICS 7:COLOR 1
30 FOR X=0 TO 159
40 FOR Y=0 TO 79
50 PLOT X,Y
60 NEXT Y
70 NEXT X
```

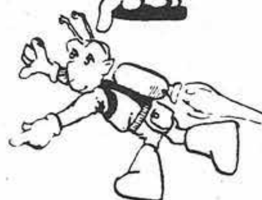
לתשומת לבך: משפט NEXT Y מופיע לפני המשפט NEXT X . הסיבה היא שלולאת Y חייבת להסתיים לפני שישתנה הערך של NEXT X .
לפיכך, כל זמן ש X=0 לולאה זו משנה את הערך של Y מ- 1,2,3,4 ועד 79. היא מבצעת את שורה 50 שמונים פעם לפני שערכו של X משתנה ל-1; ואז היא מבצעת את השורה 80 פעמים נוספות לפני הערך הבא של X. בסך הכל, מבוצעת שורה 50 12,800 פעם בזמן ביצוע שתי הלולאות ... ובמהירות רבה מאד!

☆ **לולאת "פסק זמן"** - דחיה או הפסקה, היא לולאת FOR-NEXT ריקה. היא גורמת למחשב לספור בשקט לעצמו בלי לעשות שום דבר אחר. אפשר לפקח על משך ההפסקה ע"י שינוי מספר המחזורים הלא פעילים שהמחשב עובר דרכם.
הנה לולאה איטית:



```
10 FOR WAIT=1 TO 2000
20 NEXT WAIT
```

וזו מהירה יותר :



```
10 FOR WAIT=1 TO 200
20 NEXT WAIT
```

מילים ב- BASIC

☆ STEP - הפונקציה STEP יוצרת קפיצות קבועות בערכים מספריים ומשמשת לעריכת שנויים קלים בלולאה FOR-NEXT. לפיכך :



```
10 FOR X=1 TO 5 STEP 2
20 PRINT X
```

תתחיל ב-1, אז תדלג על 2 ותגיע ל-3, תדלג שוב על ערך ותגיע ל-5. כשתרוץ, תכנית זו תדפיס :

```
1
3
5
```

תהליך זה של שינוי קבוע כלפי מעלה בערכו של משתנה נקרא גם תוספת.



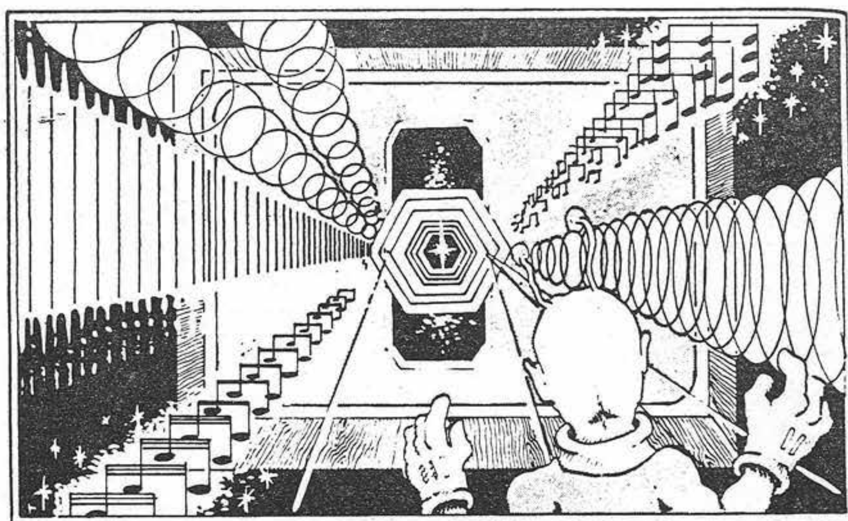
```
10 FOR X=1 TO 10 STEP 5
20 PRINT X
30 NEXT X
```

תכנית זו תדפיס : 1
6

מאחר ש-1 הוא המספר הראשון, ו- 6=1+5. לא יודפסו מספרים אחרים, מאחר שב 5 STEP 6 מגיעים ל-11, ו-11 נמצא מחוץ לתחום ערכי ה-X שלגביו הערך הגבוה ביותר עפ"י שורה 10 בתכנית הוא 10.

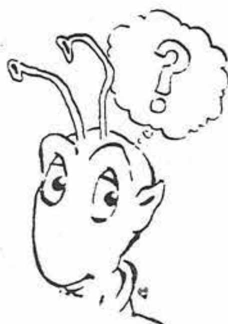
מקשים מיוחדים

☆ (CLEAR)(ESC) מקש (ESC) מאפשר להעביר פקודות ששימושן הרגיל הוא במצב ביצוע מיידי אל מצב ביצוע נדחה - שבו מבוצעות הפקודות בתוך תכנית. כדי לנקות את המסך תוך כדי כתיבת התכנית בצע את הפעולות הבאות אחרי תקתוק מספר השורה : לחץ על מקש (ESC) פעם אחת. לחץ והחזק את מקש (CTRL) ובאותו זמן לחץ על (CLEAR). מהלך זה יופיע בהדפסי תכניות בספר בצורת (CLEAR) (ESC).



עיטור תכניות משחקים

האם ראית פעם את לוחות התוצאות האלקטרוניים הנמצאים באיצטדיוני ספורט ? יש מקומות,שבהם ברגע שבועטים לשער לוח התוצאות ממש מתפוצץ - מהבהב את שמו של השחקן בתריסר צבעים, מפעיל צופרים ומקרין סרטים מצוירים. המשחק מעוטר ע"י התצוגה על המסך של לוח התוצאות. באותו אופן תוכל לעטר את המשחקים שתתכנת במחשב אטארי. ככל הנראה, אין אפקט מיוחד שלוחות התוצאות המשוכללים האלה מסוגלים להפיק, שלא תוכל לתכנת במחשב. המגבלות היחידות העומדות בפניך הן גודלו של מסך הטלביזיה ועצמתם של הרמקולים שלו.



בפרק זה נציג בפניך כמה דרכים לעטר את תכניות המשחקים שלך : למשל, תוספות מפוארות לסיום, הצגת המשחק בצורות מעניינות באופן מיוחד, השמעת תשואות למנצחים ועידוד המפסידים לנסות שנית. הגוון והעניין שיצרו המשחקים יוגברו על ידי הצורה שתעטר אותם.

ההצעות שנעלה כאן הן רק כפתיחה. לך ניתן להמציא את ההתחלות והסיומים שלך למשחקים.. אפשר לנסות תריסר מהם לאותו המשחק ולבדוק את השפעתם על חברים כדי למצוא את אלה המוסיפים את המידה הגדולה ביותר של שעשוע וסגנון למשחק.

אפשר להתחיל עם העיטור הבא. זוהי הודעת עידוד למנצח, המודפסת באותיות גדולות. כדי לקבל טקסט

מוגדל על האטארי אפשר להשתמש בכל אחד משני המצבים הגרפיים 1 GRAPHICS או GRAPHICS 2 ׳GRAPHICS 1 מדפיס אותיות בגובה הסטנדרטי וברוחב כפול.
GRAPHICS 2 מדפיס טקסט בגובה וברוחב כפולים. כדי להדפיס טקסט במצבים גרפיים אלה, עלינו להוסיף משהו לפקודת PRINT הרגילה. במקום " PRINT "TEXT משתמשים ב "PRINT #6;"TEXT.
הסיבה היא שאנחנו רוצים להדפיס באמצעות סוג מיוחד של מסך גרפי או "התקן", המוכר למחשב אטארי כהתקן מספר 6. נסה את הדוגמה הבאה :

```
10 GRAPHICS 1
20 PRINT #6; "HELLO"
```



המלה " HELLO " תודפס בפינה השמאלית העליונה של המסך. מחשב אטארי יכול גם להדפיס אותיות דפוס רגילות ("קטנות") אבל בצורה מוגדלת, אבל הדבר דורש מושגים קצת יותר מתקדמים, ואנו לא נציג אותם כעת.
אם נתקתק את הטקסט באותיות קטנות ע"י לחיצה על המקש (CAPS/LOWR), הן תופענה על המסך כאותיות גדולות אבל בצבע שונה במצבים הגרפיים 1 ו-2. ננסה את השורה הבאה :

```
PRINT #6 ; "hello"
```

יש ביכלתנו לציין במדויק את המקום שבו אנו רוצים שיודפס הטקסט - באמצעות משפט POSITION .
במצב גרפי 1 (GRAPHICS 1) עומדים לרשותנו 20 מקומות בקו אופקי (ממוספרים 0-19) ו-20 מקומות בקו אנכי, לא כוללים את חלון הטקסט בתחתית המסך (וממוספרים 0-19).

לפניך קטע אחרון ופשוט של תכנית, אשר מברך את המנצח במשחק. תוך שימוש בדרך המקובלת שהצגנו בפרק זה למיספור השורות, ה"עיטור" הזה יתחיל בשורה 2000.



```
2000 REM * CONGRATULATIONS
2002 REM *  TO THE WINNER
2004 REM *  GRAPHICS 2+16
2006 REM *
2010 GRAPHICS 2+16
2020 POSITION 6,5
2030 PRINT #6;"YOU WON!!"
2040 FOR WAIT=1 TO 1000:NEXT WAIT
```

נזכור לחזור תמיד למצב הרגיל של לוח המקשים ע"י לחיצה על (SHIFT) ו-(CAPS/LOWR)אחרי שגמרנו לתקתק אותיות קטנות !

בשורה 2010 שונה GRAPHICS 1 ל- GRAPHICS 2 (לאותיות גדולות במיוחד) ונוסף מספר 16. דבר זה נותן לנו מסך שלם במצב GRAPHICS2, בלי ה"חלון" הקטן בתחתית. אפשר לעשות אותו הדבר עם GRAPHICS 1 : שנה את שורה 2010 ל- GRAPHICS 1 + 16 והרץ את הקטע הזה שנית. כיצד הם נבדלים זה מזה ?



בשורה 2020 ממקמת הפקודה POSITION את תחילת ההודעה בנקודה מסוימת על המסך. היא משתמשת באותו רעיון כמו PLOT X,Y. X הוא מספר העמודה, Y הוא מספר השורה, והנקודה 0,0 נמצאת בפינה השמאלית העליונה של המסך. ב-GRAPHICS 2, שורה 2020 אומרת למחשב להתחיל בנקודה הנמצאת 6 רווחים של גודל אות מהשולים השמאליים ו-5 רווחים מתחת לראש המסך.

להלן הסיכום של הערכים המותרים בפקודת POSITION לאותיות גדולות ב GRAPHICS 1 ו- GRAPHICS 2.



GRAPHICS MODE	*X*	*Y*
GRAPHICS 1	0 - 19	0 - 19
GRAPHICS 1+16 (מסך מלא)	0 - 19	0 - 23
GRAPHICS 2	0 - 19	0 - 9
GRAPHICS 2+16 (מסך מלא)	0 - 19	0 - 11

דרך נוספת להחיות קצת את הודעת הנצחון היא להשתמש בפקודות BASIC של אטארי היוצרות קול. הנה כאן קטע אחרון קצר המשתמש בטקסט בגודל רגיל, יחד עם חיזוק מוסיקלי. כדאי להגדיל את עוצמת הקול של מסך הטלביזיה!

```
2100 REM * CONGRATULATIONS
2102 REM *   WITH SOUND
2104 REM *
2110 FOR I=1 TO 100
```

ראה בפרק השלישי, פרק ז', את התיאור המפורט של הפקודה SOUND X,Y,Z,W.



```
2115 POSITION INT(RND(1)*30),INT(RND(1)*23)
2120 PRINT "YOU WON!";
2130 SOUND 0,RND(1)*250,10,8
2140 NEXT I
```

הבה ננסה לעטר קצת את תחילת התכנית. נתחיל בכותרת בראשית המשחק. אפשר להשתמש בתכנית הזאת כהצגה פשוטה של המשחק למשתמש בו. שם התכנית מוצג על המסך באותיות בגודל כפול (שורות 20 ו-30) עד שלוחצים על (START) והתכנית ממשיכה.הפקודה PEEK(53279)  בשורה 100 היא פקודה שאומרת למחשב לבדוק ולראות אם ילחצו על איזשהו מקש מן המקשים המיוחדים, (START), (SELECT), או (OPTION)。
למקש (START) יש ערך של 6. אם לא לחצו על המקש הזה, חוזר המחשב ובודק שוב עד שלוחצים עליו (על (START)).

תקתק NEW ואז את התכנית :



```
10 REM * TITLE PAGE FOR PROGRAM
12 REM *
18 REM * DOUBLE SIZE TEXT WITH WINDOW
20 GRAPHICS 2
28 REM * SET CURSOR AT COLUMN 3,LINE 4
30 POSITION 3,4
38 REM * PRINT PROGRAM NAME IN
40 PRINT #6;"PROGRAM NAME"
98 REM * WAIT UNTIL START IS HIT
100 IF PEEK(53279)<>6 THEN GOTO 100
198 REM * PROGRAM NOW STARTS HERE
200 PRINT "THANK YOU."
```

יתכן שידוע לך על השמוש ב-(START), אבל שחקן אחר איננו יודע. אפשר להסביר לאחרים ע"י הוספת שורה בתכנית. יש ללחוץ על (SYSTEM RESET): נקבל הדפס של התכנית ע"י LIST ונוסיף את השורה :

```
50 PRINT "PRESS START KEY TO BEGIN"
```

הבה ננסה לראות דרכים שונות לעיטור השער של התכנית. ראשית, נדאג לכך שרקע המסך והחלון הגרפי יהיו באותו הצבע (שחור).

פקודת PEEK(X) קצת יותר מסובכת מהפקודות שנסביר בספר זה למתחילים. אבל אם אתה מכיר ומבין כבר את PEEK(X) תוכל למצוא כמה מספרים שמושיים לפקודת PEEK לתכנות משחקים בנספח III.

פשוט נוסיף את השורה הבאה :

```
25 SETCOLOR 2,0,0:REM TEXT WINDOW BLACK
```



כעת נוכל לשנות את צבע האותיות של שם המשחק כך שיהבהבו בצבעים שונים. זה פשוט מאד. עלינו רק להוסיף את שתי השורות הללו :

```
52 SETCOLOR 0,HUE,6: REM COLOR REGISTER
   CONTROLING THE LETTERS OF NAME
60 HUE=HUE+1
```

גם את שורה 100 צריך לשנות :

```
100 IF PEEK(53279)< >6 THEN
```

האפקט ממש נחמד. אם נרצה שהצבעים ישתנו בקצב איטי יותר ; פשוט נוסיף לולאת הפסקה כמו זו :

```
90 FOR WAIT=1 TO 200:NEXT WAIT
```

אין צורך לשנות את צבע האותיות. במקום זה, נוכל לשנות את צבע הרקע. יש רק לשנות את ה-0 בשורה 52 ל-4, והשורה תראה כך :

```
52 SETCOLOR 4,HUE,6:REM BACKGROUND COLOR
```

החלון הגרפי חזר.
אבל חכו , יש עוד. מה דעתך על כך שהשער יכבה לחצי שניה לפני שישנה את צבעו ? גם את זה קל לסדר !
פשוט נשנה את שורה 52 לשורה 55 ונוסיף שתי שורות חדשות 52 ו-53. נחזיר את שורה 55 לצורתה המקורית, והנה מה שיתקבל :

```
52 SETCOLOR 0,0,0:REM * BLANK OUT THE
   TITLE
53 FOR WAIT=1 TO 100:NEXT WAIT:REM *
  THIS IS A DELAY LOOP
55 SETCOLOR 0,HUE,6:REM * PUT COLOR
  BACK INTO TITLE
```

הסבר מפורט של הפקודה SETCOLOR X,Y,Z בא בחלק השלישי, פרק ח'.

HUE, כמו WAIT, הוא רק שם של משתנה, כמו X,Y או A. ב-BASIC של אטארי מותר להשתמש בשמות ארוכים למשתנים (עד 255 אותיות, אך בלי רווחים), כך שניתן לקרוא ולהבין את התכניות ביתר קלות.

ומה דעתך להעמיד את הכותרת על ראשה בכל הבהוב שני ?
זה רק נשמע מסובך, אבל הכל הוא רק ענין של פקודה אחת -
POKE.△ POKE 755,4 הופך את כל האותיות על המסך
ומעמיד אותן הפוך על ראשן. POKE 755,2 מחזיר אותן
למצב הרגיל. הוסף את השורות האלה כדי לקבל את האפקט
המיוחד הזה :

```
45 FLIP=2:REM * SET INITIAL VALUE FOR
   'FLIP' VARIABLE TO UPSIDE DOWN
46 REM * NOTE: 'FLIP=2' IS SHORT FOR
   'LET FLIP=2'
70 POKE 755,FLIP:FLIP=FLIP-6:REM *
   NOW FLIP TITLE UP AND DOWN
101 POKE 755,2:REM * RETURN CHARACTERS
   TO NORMAL BEFORE STARTING PROGRAM
```



אם היפוך האותיות בצורה הזו הוא קצת יותר מדי עבורך,
מה דעתך על שינוי האותיות מגדולות לקטנות ואז
חזרה לגדולות ? זה קל באותה מידה כמו להעמיד אותן
על הראש - צריך רק להוסיף :

```
45 FLIP=224: REM INITIALIZE VARIABLE FOR
   FLIP (POKE)
70 POKE 756,FLIP:FLIP=450-FLIP:REM FLIP
   TITLE UP & DOWN
101 POKE 756,224:REM RETURN CHARACTERS
   TO NORMAL UPPER CASE MODE
```

השנוי הזה יתן לנו כותרת מתחלפת בין אותיות גדולות
וקטנות. אבל, בדרך כלל מתלוה לזה תופעה מעניינת
ובדרך כלל בלתי רצויה : כל פעם, שהכותרת מופיעה
באותיות קטנות, מופיעים גם לבבות צבעוניים !
אם זה מוצא חן בעיניך, הרי שאין כל בעיה. אבל אם
לא, אין לדאבוננו דרך פשוטה להפטר מהלבבות, והסיבה
היא האופן שבו מסודרים הצופנים של האותיות במחשב
אטארי. אנו יכולים להשתמש בתכונה זו כדי להבין

△ POKE היא פקודה מיוחדת שמכניסה ערך מספרי ישירות
לתוך מקום מסוים בזכרון של מחשב אטארי. היא דומה
לפקודת BASIC: משפט =LET X. אבל POKE יכולה
לעשות דברים שאי אפשר לעשות באמצעות BASIC , כמו למשל
העמדת אותיות על ראשן. מידע נוסף על POKE והפקודה
ההפוכה שלה, ____ , תמצא בנספח ובמדריך של אטארי לשפת
BASIC. צריך להשתמש ב POKE בזהירות, משום שהיא
עלולה לגרום לשנויים בלתי רצויים בתכנית אם לא נבין
בדיוק מה אנחנו עושים.

קצת יותר את שנויי הצבעים. לא נסביר כאן במדויק כיצד השנויים האלה פועלים, אבל אולי נרצה להשתמש בשנויים הקלים בתכנית שבשורות הבאות, ולהסתכל בפרק ח' בדוגמאות נוספות שמראות כמה צבעוני יכול להיות מחשב האטארי שלך !
צריך לתקתק את שורה 40 מחדש, כך :

40 PRINT #6; "program name"

כדי לקבל אותיות קטנות, נלחץ על (CAPS/LOWER) פעם אחת לפני תקתוק האותיות. כדי לחזור לאותיות גדולות, נלחץ ונחזיק אחד ממקשי (SHIFT) ובאותה עת נלחץ שוב על (CAPS/LOWER) . מקש זה הוא כמו מפסק ; הוא מדליק ומכבה אותיות גדולות/קטנות.

עכשיו, הרץ את התכנית. מה קרה ? הלבבות עדיין ישנם חצי מהזמן, אבל הכותרת תמיד באותו צבע.
תקתוק הכותרת באותיות קטנות בתכנית לא שינה את האותיות לקטנות אחרי הריצה.

נשנה את שורה 55 : 55 SETCOLOR 1,HUE,6

זה גורם לשנוי הצבע בכותרת. נתקתק גם את השורה הבאה ונריץ את התכנית מחדש. מה קורה ? 52 SETCOLOR 0,0,0

האם התכנית פעלה בצורה שחשבת שתפעל ? או שמא הלכת קצת לאיבוד באמצע הדרך ?
בתכנית הבאה צירפנו רבות מן השורות שהופיעו בדפים האחרונים כך שנוכל להשתמש בהן כדי ליצור שער מעניין לאחת התכניות שלנו. נוכל לנסות אותה עכשו או לשוב אליה מאוחר יותר.

```
10 REM * TITLE PAGE FOR YOUR PROGRAM
12 REM *
18 REM * DOUBLE-SIZE TEXT WITH WINDOW
20 GRAPHICS 2
24 REM * SET BACKGROUND COLOR TO BLACK
25 SETCOLOR 2,0,0
30 POSITION 3,4
40 PRINT #6;"PROGRAM NAME"
45 FLIP=224
50 PRINT "PRESS START KEY TO BEGIN"
52 SETCOLOR 0,0,0
53 FOR WAIT=1 TO 200:NEXT WAIT
55 SETCOLOR 0,HUE,6:REM * COLOR TITLE
```

```
60 HUE=HUE+1
70 POKE 756,FLIP:FLIP=450-FLIP
90 FOR WAIT=1 TO 200:NEXT WAIT
100 IF PEEK(53279)<>6 THEN GOTO 52
101 POKE 756,224
200 REM * PROGRAM THEN BEGINS HERE
```

העניינים מסתבכים

בנקודה זו ברצוננו להראות לך כמה דרכים מורכבות באמת לעטר את התכניות שלך. אבל כדי לעשות זאת עלינו להכנס קצת לאפיונים מיוחדים של כמה תכונות מורכבות של מחשב אטארי, כמו : תופעת "inverse"
שבה מופיעות האותיות כהות על רקע לבן במקום הצורה הרגילה של אותיות לבנות על רקע כהה; סימנים גרפיים; ואותיות בקרה שמאפשרות לך להדפיס בכל מקום על המסך (למעלה,למטה,משמאל או מימין). חברת CODE WORKS™, המוציאים לאור של IRIDIS™ יצרו תכניות ותיעוד מצוינים למחשב אטארי. הם פיתחו שיטת אפיון מיוחדת שמטפלת בתכונות האלה, ובאדיבותם התירו לנו להשתמש בשיטתם לתכניות בספרנו.

לצורת האפיון יש שני חוקים יסודיים : כל דבר שמתחתיו מתוח קו יופיע באותיות "inverse" וכל דבר בסוגריים הוא תו מיוחד. השאר נראה כפי שהוא מופיע.

תו או סימן יחיד או סימן פיסוק בסוגריים מיצגים אות גרפית. תקתק את התוים האלה ע"י לחיצה והחזקה של מקש הבקרה (CTRL) תוך כדי לחיצה על המקש המצוין בתוך הסוגריים. למשל (S) הוא S בקרה ו-(.) הוא נקודת בקרה.

S בקרה יופיע על המסך כצלב ; נקודת בקרה תופיע כמעוין. כשהכוונה היא להדפיס את התו הגרפי ב-inverse, יופיע קו מתחת לתו שבתוך הסוגריים. כך, (S) מיצג את הלב הגרפי ב inverse.

מלה בתוך סוגריים מציינת פונקצית בקרה מיוחדת. (CLEAR) ינקה את המסך. (BELL) מזמזם. כדי להכניס את התו של (CLEAR) עליך ללחוץ (ESC) (מן המלה ESCAPE ,לברוח)פעם אחת ואז להחזיק את (SHIFT) תוך כדי לחיצה על מקש (CLEAR). על המסך יראה התו של (CLEAR) כחץ מעוקל המצביע שמאלה. כדי להכניס את התו של (BELL) לחץ שוב על (ESC), החזק את (CTRL) ותוך כדי כך לחץ על מס' 2. הטבלה הבאה מסבירה כיצד לתקתק כדי לקבל את כל התוים של הפונקציות במקשים המיוחדים.

התכניות שהדפסן מופיע בספר שונות מאלו של CODE WORDS™

אם אתה מעונין לדעת יותר על IRIDIS™ זו הכתובת :

The Code Works™, P.O.Box 550
Goleta, California 93017

רק בצורה אחת : כאשר מתקתקים את התו המיוחד יותר מפעם אחת, בזה אחר זה. תכניות CODEWORKS TM משתמשות ב (,)3 כדי ליצג 3 תוים של "לבבות" גרפיים. בספר זה ניצג שלשה "לבבות" כך :

.(,) (,) (,)

לסיכום, יש שתי דרכים להשיג את הצורות של הגרפיקה האוטומטית שבנויה אל תוך מחשב אטארי :

1) ללחוץ על (CTRL) ולהחזיק, ותוך כדי כך ללחוץ על המקש המצויין בסוגריים. הצורה הגרפית תודפס על המסך כל זמן שתחזיק את מקש (CTRL).

2) אם ברצונך לתקתק את הצורות הגרפיות בלי להיות צמוד למקש CTRL , הנה מה שעליך לעשות :
לחץ (CTRL) ועל CAPS/LOWR , ביחד. אח"כ לחץ על המקש המצויין בסוגריים (אחרי שעזבת את CTRL ו- CAPS/LOWR). כדי להשתחרר ממצב זה ולחזור לאותיות גדולות לחץ על (SHIFT) ועל CAPS/LOWR ביחד.

וכעת הנה תכנית שמשתמשת בכמה מהתכונות המיוחדות האלה. שים לב שבשורה 30 מופיעה המילה START עם אותיות מודגשות בקו. פירוש הדבר הוא, שצריך ללחוץ על המקש עם הסמל של אטארי ושהאותיות המודגשות תופענה על המסך במצב inverse (להדגשה). זכור ללחוץ על מקש ה inverse שוב אחרי תקתוק המלה, כדי לחזור למצב אותיות רגיל.

```
10 REM * TITLE DISPLAY
12 REM *
20 GRAPHICS 2:REM * LARGE TEXT
30 PRINT "PRESS START TO BEGIN":REM *
  THIS IS PRINTED IN THE SMALL TEXT
  WINDOW AT BOTTOM OF SCREEN
40 POSITION 3,4:PRINT #6;"PROGRAM NAME"
98 REM * CHECK TO SEE IF START
100 IF PEEK(53279)<>6 THEN GOTO 40
198 REM * PROGRAM BEGINS NOW
200 PRINT "THANK YOU."
```

ראשית, הבה נוסיף שורה 25 כדי לקבוע צבע שחור לחלון, כמו לרקע של שאר המסך :

```
25 SETCOLOR 2,0,0:REM TEXT WINDOW BLACK
```

על המסך	מקש	אתה מתקתק
	{A}	ctrl-A
	{B}	ctrl-B
	{C}	ctrl-C
	{D}	ctrl-D
	{E}	ctrl-E
	{F}	ctrl-F
	{G}	ctrl-G
	{H}	ctrl-H
	{I}	ctrl-I
	{J}	ctrl-J
	{K}	ctrl-K
	{L}	ctrl-L
	{M}	ctrl-M
	{N}	ctrl-N
	{O}	ctrl-O
	{P}	ctrl-P
	{Q}	ctrl-Q
	{R}	ctrl-R
	{S}	ctrl-S
	{T}	ctrl-T
	{U}	ctrl-U
	{V}	ctrl-V
	{W}	ctrl-W
	{X}	ctrl-X
	{Y}	ctrl-Y
	{Z}	ctrl-Z

על המסך	מקש	אתה מתקתק
	{,}	ctrl-comma
	{.}	ctrl-period
	{;}	ctrl-semicolon
	{BACK}	ESC BACK
	{BELL}	ESC ctrl-2
	{CLEAR}	ESC shift-less or ESC ctrl-less
	{CLR TAB}	ESC ctrl-TAB
	{DEL CHAR}	ESC ctrl-BACK
	{DEL LINE}	ESC shift-BACK
	{DOWN}	ESC ctrl-equals
	{ESC}	ESC ESC
	{INS CHAR}	ESC ctrl-greater
	{INS LINE}	ESC shift-greater
	{LEFT}	ESC ctrl-plus
	{RIGHT}	ESC ctrl-star
	{SET TAB}	ESC shift-TAB
	{TAB}	ESC TAB
	{UP}	ESC ctrl-minus

---- Note ----

```
comma     is ,
equals    is =
greater   is >
less      is <
minus     is -
period    is .
plus      is +
semicolon is ;
star      is *
```

סדר תווי אטארי על-פי
שיטת IRIDIS.

כעת, ניתן לכותרת להתחיל בראש המסך ואז לרדת על פני המסך. שנה את שורה 100 ל 29 THEN GOTO ושנה את ה 4 בשורה 40 ל ROW, וזה מה שאתה צריך לקבל :

```
40 POSITION 3,ROW:PRINT #6;"PROGRAM NAME";
100 IF PEEK(53279)< >6 THEN GOTO 29
```

הוסף שורות אלה

```
29 PRINT #6;"[CLEAR]";:REM CLEAR
   SCREEN
50 ROW=ROW+1:IF ROW> 9 THEN ROW=0
```

נזכור ש (CLEAR) בשורה 29 משמעותה שאפשר ללחוץ על (ESC) ואז על (CTRL) ו- (CLEAR) ביחד.
זה מאפשר לחלון להתנקות תוך כדי הרצת התכנית.
כותרת המשחק שלך רצה לאורך המסך, קרוב לודאי מהר יותר משהיית רוצה. לכן נוסיף הפסקה :

```
90 FOR WAIT=1 TO 200:NEXT WAIT
```

ע"י הוספת השורה הבאה יכול "דף השער" שלך להחליף צבעים תוך כדי שהוא נע לאורך המסך :

```
80 SETCOLOR 0,0,ROW*17+238:REM CHANGE
   THE COLOR OF TITLE
```

נשים לב שאנו משנים את הצבע ע"י שנוי רמת הבהירות.
נראה, שרמת בהירות מעל 14 תשפיע גם על הצבע.
האפקט משביע רצון ותוכל לנצל אותו. ועוד אפקט אחד נוסף שאולי כדאי לראות בפעולה. נוסיף שורה 1 לכל אחת מן התכניות הנ"ל :

```
1 POKE 77,128:REM SCREEN PROTECTION ON
```



עכשו יתחיל לפעול מנגנון שנוי צבע המסך הבנוי אל תוך המחשב. מענין, מה ? או ננסה להשתמש בערך 126 במקום 128 לכמה שניות השהיה לפני שהוא מתחיל.

זהו זה לגבי תיבול מענין של "דף שער" פשוט. התנסה בשימוש במידע בדפים האחרונים, ערכו כמה שינויים, בדקו את האפשרויות-ומעכשו, הענינים בידיכם.

פרק ג'

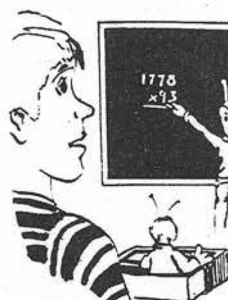


תרגול ומיומנויות יסוד

התרגול הוא טרחה משעממת ! פעולות החשבון,טבלאות, רשימות מלים ומרכיבים מכניים אחרים של הלמידה דורשים חזרה ולימוד בעל פה. התענוג היחידי הטמון בהם נובע מכך שמשתלטים עליהם ולא צריך לתרגל אותם שוב. באמצעות מחשב אטארי, אפשר לבנות תכניות תרגול ולעטר אותן כך שהתרגול יהיה יותר מעניין.

יש תכניות שינון ותרגול שאפשר להתאימן לצרכים המיוחדים של ילדים. החלק הראשון בפרק זה ידון במספרים :חיבור, חיסור, כפל, חילוק ואלגברה. חלקו השני של הפרק יכיל כמה תכניות פשוטות שיהיו לעזר בכתיב, אוצר מלים ודקדוק.

נתחיל בכמה תכניות המשתמשות באוצר המלים של BASIC שכבר הצגנו ואז נראה כיצד אפשר לשנות אותן מעט כדי לעזור לילדים בפעולות המכניות של קריאה, כתיבה וחשבון. ואז נציג כמה תכניות יותר מתוחכמות המשתמשות במשחקים ואשר מספקות אותו סוג תרגול.



פעולות החשבון

התכנית הזו צריכה לענין ילדים בני חמש עד שבע. המחשב מציג מספר על המסך ומבקש מן השחקן לתת (ע"י INPUT) את המספר הבא. אם השחקן מנחש נכון את המספר, נותן המחשב לנחש מספר נוסף. אם לא, מדפיס המחשב "נסה שנית" ונותן לשחקן אפשרות נוספת לנחש. בתכנית זו, כמו בתכניות טובות אחרות שבהן יש קשר ישיר בין המחשב למשתמש, מדפיס המחשב הודעה כלשהי לפני שמתקתקים את התשובה. כך, בשורה 120 מדפיס המחשב

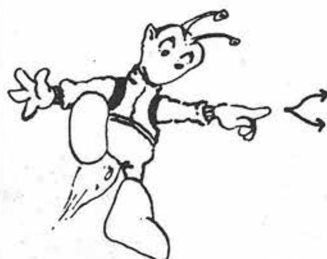
"WHAT NUMBER COMES RIGHT AFTER ____________"

ובמקום הקו צריך לבוא הערך של X. המשתנה X מקבל את ערכו דרך פונקצית RND שתייצר לו מספרים סתמיים, אקראיים, המופיעה בשורה 110. המספר שנקבל יהיה מספר אקראי בין 0-10 . כשתראה "?" מופיע על המסך, פירוש הדבר הוא שהמחשב מחכה לקבל את תשובתך.

זכור לכלול במשחקים שלך "רמזים" שיאמרו לשחקן מה עליו לעשות.

הרץ את התכנית הפשוטה הזו. היא תמשיך לרוץ עד שילחצו על (BREAK) או על (SYSTEM RESET).

זוהי גירסת "שלד" שתתאים לסוגים רבים של תכניות המטפלות בפעולות החשבון או ניחוש מספר לא ידוע.



```
10 REM * FIRST DRILL FOR LITTLE ONES
12 REM *  (C) 1980 BY HERBERT KOHL
14 REM * (no dress-up)
16 REM *
100 PRINT "[CLEAR]":REM * CLEAR SCREEN
105 REM * CHOOSE RANDOM NUMBER 0 - 10
110 LET X=INT(RND(1)*11)
120 PRINT "WHAT NUMBER COMES RIGHT AFTER
 ";
130 PRINT X;
200 REM *
202 REM * GET THE ANSWER
204 REM *
210 INPUT Y
220 IF Y=X+1 THEN GOTO 2000:REM * CORREC
T ANSWER
230 GOTO 3000:REM * INCORRECT ANSWER
2000 REM *
2002 REM * MESSAGE FOR CORRECT ANSWER
2004 REM *
2010 PRINT :PRINT "GOOD"
2020 PRINT :PRINT
2030 GOTO 110
3000 REM *
3002 REM * MESSAGE FOR INCORRECT ANSWER
3004 REM *
3010 PRINT :PRINT "TRY AGAIN..."
3020 PRINT
3030 GOTO 120
```

אפשר לעטר תכנית פשוטה זו בכמה דרכים שונות. אפשר להוסיף הסתעפות שתאפשר לשחקן להחליט אחרי כל ניחוש האם הוא רוצה להמשיך לשחק או להפסיק. או אפשר להשתמש בצבע ובאותיות ומספרים גדולים על המסך שיקלו על הקריאה על ילדים יותר קטנים. הגירסה הבאה של המשחק כוללת את השינויים האלה ומשתמשת בכמה מלים חדשות ב BASIC. הכרנו כבר את GRAPHICS 2 (אותיות גדולות) ו- # PRINT בפרק ב'. נסביר את המושגים DIM A$, INPUT A$, GRAPHICS 0 ו- END בהמשך

הפרק. שני המושגים החשובים ב BASIC בגירסה הזו של התכנית הם GOSUB ו- RETURN.



GOSUB ו- RETURN מייצגים רעיון תכנותי רב עצמה, זה של תת-תכנית. תת-תכנית היא תכנית קטנה בתוך תכנית. למשל, בשנוי שערכנו בתכנית התרגול שנביא מיד, אנו משתמשים בארבע תת-תכניות. הן נמצאות בשורות 2000 עד 2099, 3000 עד 3099, 4000 עד 4060 ו-5000 עד 5099. פקודת GOSUB הראשונה בשורה 320 הולכת לתת התכנית המתחילה בשורה 2000. משתמשים בה כדי להדפיס את ההודעה "GOOD" עבור תשובה נכונה. אח"כ, הפקודה חוזרת (RETURN) לשורה 330 וממשיכה בתכנית. שורה 330 גם היא פקודת GOSUB ; הפעם היא משתמשת בתת התכנית בשורה 5000.
תת-התכנית גורמת למחשב השהיה קצרה אחרי הדפסת ההודעה. שורה 400 פותחת סדרה של פקודות GOSUB שמשתמשים בהן אם התשובה אינה נכונה : קודם בתת התכנית בשורה 3000 ואז בהשהיה בשורה 5000. שים לב שאפשר לעבור ע"י GOSUB מתת תכנית אחת לאחרת. שתי תת התכניות בשורות 2000 ו-3000 "קוראות" לתת התכנית בשורה 5000 לבצע השהיה קצרה לפני שהן חוזרות לתכנית הראשית.

```
10 REM * SIMPLE DRILL FOR LITTLE ONES
12 REM *      COPYRIGHT 1981 BY
14 REM *  HERB KOHL AND TED M. KAHN
16 REM * uses large letters & numbers
20 REM *
50 DIM A$(1)
70 REM * PROBLEMS IN GROUPS OF 10
100 FOR I=1 TO 10
110 REM * CHOOSE NUMBER FROM 0 - 10
120 LET X=INT(RND(1)*11)
200 REM * USE GRAPHICS 2 TEXT
210 GRAPHICS 2
220 PRINT #6;"[CLEAR]WHAT NUMBER COMES"
230 PRINT #6;"RIGHT AFTER ";X;"?"
```

```
300 REM * GET THE ANSWER
310 INPUT Y:PRINT #6;Y
315 REM * IF CORRECT, THEN DO ANOTHER
320 IF Y=X+1 THEN GOSUB 2000:GOTO 340
325 REM * OTHERWISE, TRY AGAIN...
330 GOSUB 3000:GOTO 210
340 NEXT I
400 REM * ASK TO PLAY AGAIN?
410 GOSUB 4000
420 GOTO 100
2000 REM *
2010 REM * CORRECT ANSWER MESSAGE
2020 REM *
2030 PRINT #6:REM * PRINT BLANK LINE
2040 PRINT #6;"GOOD!"
2050 GOSUB 5000:REM * PAUSE BRIEFLY
2099 RETURN
3000 REM *
3010 REM * INCORRECT ANSWER MESSAGE
3020 REM *
3030 PRINT #6:REM * PRINT BLANK LINE
3040 PRINT #6;"NO, TRY AGAIN..."
3050 GOSUB 5000:REM * PAUSE BRIEFLY
3099 RETURN
4000 REM *
4010 REM * PLAY AGAIN?
4020 REM *
4030 PRINT "[CLEAR]DO YOU WANT TO PLAY A
GAIN":PRINT "(YES OR NO)";
4040 INPUT A$:REM * IF "Y" GO BACK
4050 IF A$="Y" THEN RETURN
4060 GOTO 9999:REM * IF "N" THEN END
5000 REM *
5010 REM * PAUSE OR DELAY ROUTINE
5020 REM *
5030 FOR WAIT=1 TO 600:NEXT WAIT
5099 RETURN
8999 GRAPHICS 0:END :REM * END OF PROGRA
M
```

התכנית נראית מסובכת, אך הסבר קצר לגבי כמה משפטים בתכנית יעזור לפשט את הדברים. למעשה, זו אותה התכנית כמו בגירסה 1. שורה 320 מראה שמוש מעניין של IF THEN.

אם כותבים אחרי החלק THEN של משפט התנאי הזה יותר מפקודת BASIC אחת כשהיא מופרדת ע"י נקודותיים (:), יבצע המחשב כל פקודה לפי הסדר כאשר התנאי יתקיים. לכן, רצף הפקודות למחשב בשורה 320 הוא כזה :

1. בדוק אם (IF)   Y= X+1
2. אם כן, אז (THEN) קפוץ לשורה 2000 (GOSUB 2000); אחרי החזרה (RETURN), לך (GOTO) לשורה 340.
3. אם (IF) Y איננו שוה ל X+1 , אז (THEN) המשך עם הבטויים הבאים (שורה 325 ו-330).

באופן דומה, שורה 330 קופצת קודם לתת התכנית בשורה 3000 (GOSUB 3000). אחרי שבבצוע מגיעים ל-RETURN בשורה 3099, התכנית חוזרת וגומרת את שורה 330, כלומר, חוזרת לשורה 210.



אם הזרימה בין תת התכניות האלה ובין התכנית הראשית נראית בעיתית, כדאי לחשוב על ביטוי RETURN כעל מברק, שאומר :"שלחו אותי חזרה למקום שבו קראו לי עם הפקודה האחרונה של GOSUB". המעקב אחרי ה"זרימה" של תכנות מחשב באופן כזה הוא דרך טובה מאד להבין את מבנה התכנית.

התכניות הבאות הן דוגמאות של תרגול בחשבון ואפשר להשתמש בהן לתרגל חיבור, חיסור, כפל וחילוק.

ילדים מגיל 5 ועד 12 יוכלו להשתמש בתכניות אלה, תלוי בגודל המספרים בתכנית. למשל, תרגיל חיבור עם מספרים קטנים מ-5 מתאימים לילדי גן וכתה א'.
ע"י שינוי קל בתכנית אפשר לגרום לכך שהמחשב יבנה תרגילי חיבור עם מספרים עד 10, 1000, 2000 או כל מספר אחר. לצורך הנוחיות, ישתמשו כל התכניות שנביא כאן כדוגמה במספרים עד עשר.
אפשר לשנות את המספרים בתרגיל ע"י שינוי בשורה שעליה מצביעים טא*רי או טא*רי הקטנה.

בתכנית פשוטה זו מציג המחשב בעיה בחיבור. אם מנחשים נכון, יתן המחשב בעיה נוספת. אם לא, אפשר לנסות שוב כדי להגיע לתשובה הנכונה.

```
10 REM * VERY SIMPLE ADDITION DRILL
20 REM * (C) 1981 BY HERBERT KOHL
30 REM * (without subroutines)
```



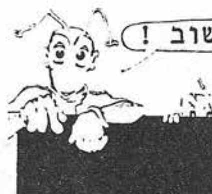
```
40 REM *
100 PRINT "[CLEAR]HOW MANY ADDITION PROB
LEMS"
105 PRINT "WOULD YOU LIKE";
110 INPUT N
120 FOR I=1 TO N
130 REM * SIZE OF NUMBERS DEPENDS ON
135 REM * VALUES IN LINES 140 & 150
140 LET X=INT(RND(1)*10)+1
150 LET Y=INT(RND(1)*10)+1
160 PRINT "WHAT IS THE ANSWER?"
170 PRINT X;" + ";Y;" = ";
200 INPUT Z
210 IF Z=X+Y THEN GOTO 300
220 PRINT :PRINT "TRY AGAIN...":PRINT
230 GOTO 170
300 PRINT :PRINT "YOU GOT IT!"
310 IF I<N THEN PRINT "HERE'S ANOTHER ON
E TO TRY"
320 PRINT
330 NEXT I
```

ע"י החלפת המלה ADDITION, חיבור בשורה 100 במילה MULTIPLICATION, כפל, והחלפת סימן חיבור (+) בסימן כפל (*) בשורות 170 ו-210, אפשר ליצור את התכנית הבאה של תרגול בכפל :

```
10 REM * SIMPLE MULTIPLICATION DRILL
20 REM * (C) 1981 BY HERBERT KOHL
30 REM * (without subroutines)
40 REM *
100 PRINT "[CLEAR]HOW MANY MULTIPLICATIO
N PROBLEMS"
105 PRINT "WOULD YOU LIKE";
110 INPUT N
120 FOR I=1 TO N
130 REM * SIZE OF NUMBERS DEPENDS ON
135 REM * VALUES IN LINES 140 & 150
140 LET X=INT(RND(1)*10)+1
150 LET Y=INT(RND(1)*10)+1
160 PRINT "WHAT IS THE ANSWER?"
170 PRINT X;" * ";Y;" = ";
```



```
200 INPUT Z
210 IF Z=X*Y THEN GOTO 300
220 PRINT :PRINT "TRY AGAIN...":PRINT
230 GOTO 170
300 PRINT :PRINT "YOU GOT IT!"
310 IF I<N THEN PRINT "HERE'S ANOTHER ON
E TO TRY"
320 PRINT
330 NEXT I
```



תרגילי חיסור דורשים שינוי בתכנית, אם רוצים שההפרש יהיה תמיד חיובי. הסיבה היא שאם המחשב יבחר ב-8 עבור X ו- 9 עבור Y, נקבל תרגיל חיסור : ? = 8-9, ובתרגיל זה ההפרש איננו מספר חיובי. אבל לא קשה לשנות את התכנית. כל מה שעלינו לעשות הוא לוודא ש- X גדול מ- Y לפני שאומרים למחשב להדפיס את התרגיל. בדוגמה הבאה הוספנו הסתעפות בשורה 155, שדואגת לענין. המשפט IF-THEN אומר שאם X קטן מ- Y, אז על התכנית לבחור זוג מספרים אחר. אם לא, עליה להדפיס את תרגיל החיסור.



```
10 REM * SIMPLE SUBTRACTION DRILL
20 REM * (C) 1981 BY HERBERT KOHL
30 REM * (without subroutines)
40 REM *
100 PRINT "[CLEAR]HOW MANY SUBTRACTION P
ROBLEMS"
105 PRINT "WOULD YOU LIKE";
110 INPUT N
120 FOR I=1 TO N
130 REM * SIZE OF NUMBERS DEPENDS ON
135 REM * VALUES IN LINES 140 & 150
140 LET X=INT(RND(1)*10)+1
150 LET Y=INT(RND(1)*10)+1
155 IF X<Y THEN GOTO 140
160 PRINT "WHAT IS THE ANSWER?"
170 PRINT X;" - ";Y;" = ";
200 INPUT Z
210 IF Z=X-Y THEN GOTO 300
220 PRINT :PRINT "TRY AGAIN...":PRINT
230 GOTO 170
300 PRINT :PRINT "YOU GOT IT!"
310 IF I<N THEN PRINT "HERE'S ANOTHER ON
E TO TRY"
320 PRINT
330 NEXT I
```

תרגילי חילוק דורשים גם הם שינוי אם ברצוננו להמנע מתוצאות שהן שברים או מחילוק ב-0. החילוק ב-0 אינו בעיה, אם משתמשים בתכנית רק במספרים מ-1 עד 10. אם שכחת כיצד פועל הדבר, חזור לפרק א', לחלק הדן בפונקציה RND(1).

על מנת לקבל תשובות לתרגילי חילוק שהן מספרים שלמים, הדרך הטובה ביותר היא לתת למחשב לבחור X ו- Y שהם מספרים שלמים, ולתת למחשב לכפול אותם זה בזה : X*Y=W.
כעת נשתמש ב- W כדי לבנות את התרגיל : W/Y = ?

נתן פקודת INPUT למספר אחר, Z ונראה :
אם Z = X אזי תוכנית החילוק נכונה.

אנו מאמצים את השיטה הזו בשורות 140 עד 155 ובשורות 170 עד 210 בתכנית תרגילי החילוק שלפניך:





```
10 REM *   SIMPLE DIVISION DRILL
20 REM * (C) 1981  BY HERBERT KOHL
30 REM *   (without subroutines)
40 REM *
100 PRINT "[CLEAR]HOW MANY DIVISION PROB
LEMS"
105 PRINT "WOULD YOU LIKE";
110 INPUT N
120 FOR I=1 TO N
130 REM * SIZE OF NUMBERS DEPENDS ON
135 REM * VALUES IN LINES 140 & 150
140 LET X=INT(RND(1)*10)+1
150 LET Y=INT(RND(1)*10)+1
155 LET W=X*Y
160 PRINT "WHAT IS THE ANSWER?"
170 PRINT W;" / ";Y;" = ";
200 INPUT Z
210 IF Z=W/Y THEN GOTO 300
220 PRINT :PRINT "TRY AGAIN...":PRINT
230 GOTO 170
300 PRINT :PRINT "YOU GOT IT!"
310 IF I<N THEN PRINT "HERE'S ANOTHER ON
E TO TRY"
320 PRINT
330 NEXT I
```



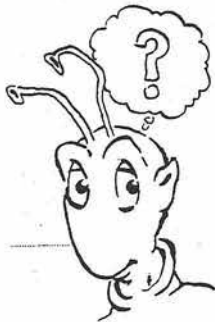
בשלב זה תוכל לכתוב תכניות תרגילים בחיבור, חיסור, כפל וחילוק עבור כל גודל מספרים שהוא. כדי לעשות את התכניות הבסיסיות האלה יותר מענינות, אפשר לעטר אותן בקול, צבע וגרפיקה. אם זה נראה מאד חשוב אפשר לתכנת את המחשב לנגן "התקוה", לפוצץ זיקוקים ולהציג קשת בשמים כל פעם שמישהו מהשחקנים מגיע ל-10 או 20 (או כל מספר אחר) של תשובות נכונות.

הנה עיטור צנוע של תכנית החיבור. הוא משתמש במילת BASIC חדשה, DIM בשורה 50 ובסוג חדש של משתנה המיוצג ע"י $ NAME בשורות 50, 70 ו-320.
שני אלה מאפשרים למחשב לטפל באותיות ובמספרים כאחד ויוסברו בפירוט בהמשך הפרק. אפשר להשתמש בהם בתכנית הזו עכשיו ולהבין כמה דברים על אופן השימוש בהם.



```
10 REM * VERY SIMPLE ADDITION DRILL
20 REM *   WITH A LITTLE DRESSING
30 REM * (C) 1981 BY HERBERT KOHL
40 REM *
50 DIM NAME$(20)
60 PRINT "[CLEAR]WHAT IS YOUR NAME";
70 INPUT NAME$
100 PRINT "[CLEAR]HOW MANY ADDITION PROB
LEMS"
105 PRINT "WOULD YOU LIKE, ";NAME$;
110 INPUT N
120 FOR I=1 TO N
130 REM * SIZE OF NUMBERS DEPENDS ON
135 REM * VALUES IN LINES 140 & 150
140 LET X=INT(RND(1)*10)+1
150 LET Y=INT(RND(1)*10)+1
160 PRINT "WHAT IS THE ANSWER?"
170 PRINT X;" + ";Y;" = ";
200 INPUT Z
210 IF Z=X+Y THEN GOTO 300
220 PRINT :PRINT "TRY AGAIN...":PRINT
230 GOTO 170
300 REM *  CONGRATULATIONS
310 FOR T=1 TO 10
320 PRINT "GREAT JOB, ";NAME$;"! ";
330 NEXT T
390 NEXT I
```

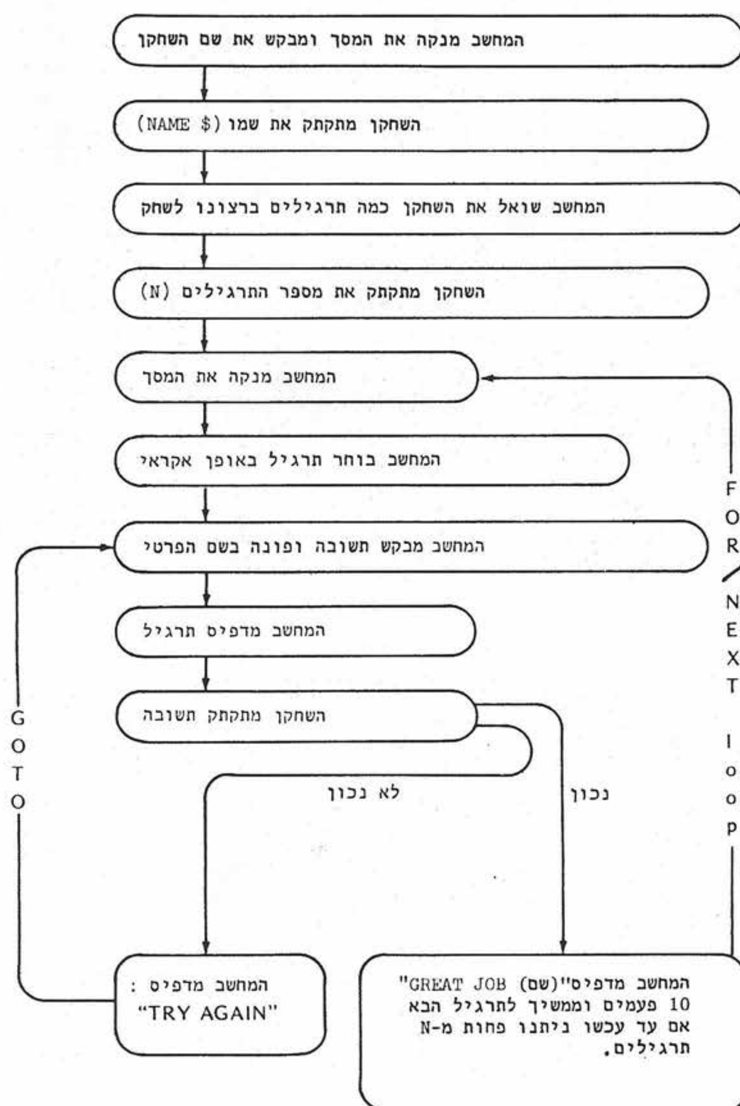




תכנית זו מבקשת מן השחקן לציין את שמו ואז מבקשת לדעת כמה תרגילים ברצונו לפתור, ושואלת זאת בפניה אישית אל השחקן. אם התשובה לתרגיל נכונה, מדפיסה התכנית "GREAT JOB,NAME $" (ובמקום NAME $ בא השם) - 10 פעמים. מבקשים מן השחקן לנסות תרגיל נוסף, עד לאותו מספר כולל של תרגילים שבקש השחקן בראשית התכנית. התכנית כמעט זהה לתכנית הפשוטה של תרגילי חיבור שהצגנו קודם, בהבדל אחד - היא נעשית אישית יותר ע"י הפניה האישית בשם ; כמו כן הוספנו קישוט קטן בשורות 300-330, כדי לברך את השחקן לכבוד פתרון נכון של התרגיל.

נתבונן במבנה התכנית הזו ובעיטורים שלה. שורה 60 מבקשת לדעת את השם הפרטי, שורות 50 ו-70 מאפשרות למחשב לאחסן את השם בזכרון באמצעות המשתנה NAME $ בכל פעם שנבקש מן המחשב להדפיס NAME $,נקבל את השם המאוחסן ב- NAME $. שורות 60 ו-125 מנקות את המסך ושורות 100 ו-160 כותבות "הודעות אישיות". לבסוף, שורות 300-360 אומרות למכשיר להדפיס את שם השחקן ליד המלים "GREAT JOB" 10 פעמים, ושורה 390 סוגרת לולאת FOR-NEXT שאומרת למחשב להציג בעיה נוספת.

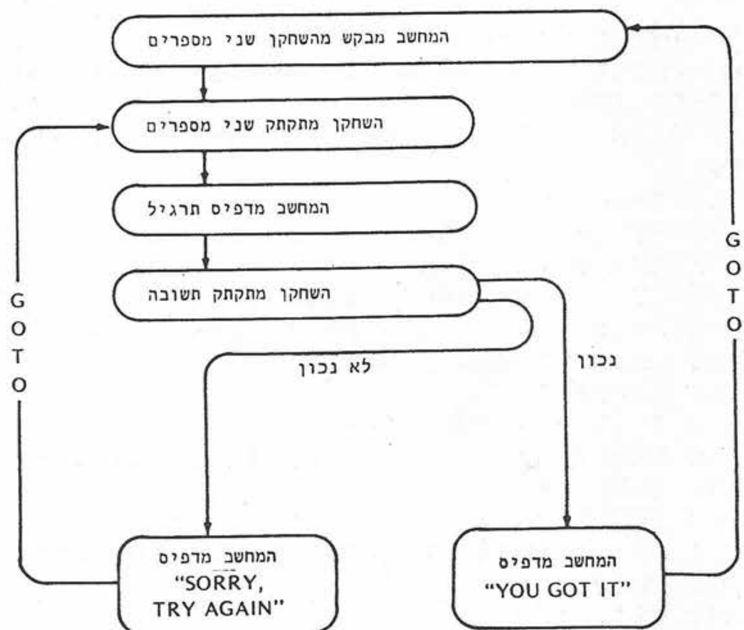
אילו רצינו לצייר דיאגרמה, או "תרשים זרימה" של התכנית הזו, הם עשויים להראות כך :



הנה עוד תכנית תרגול בחיבור שבה מחליט השחקן,
לא המחשב, על המספרים שיופיעו בתרגילים. האם
זה ברור לך באיזו שורה בתכנית זה מתרחש ?

```
10 REM *    ADDITION OF TWO NUMBERS
20 REM *         (YOU PICK THEM)
30 REM *
100 PRINT "[CLEAR]"
110 PRINT "TIME FOR ADDITION."
120 PRINT "PICK TWO NUMBERS & ADD THEM"
130 INPUT X,Y
140 PRINT X;" + ";Y;" = ";
150 INPUT Z
160 IF Z=X+Y THEN GOTO 200
170 IF Z<>X+Y THEN GOTO 300
200 PRINT :PRINT "YOU GOT IT!"
210 GOTO 120
300 PRINT :PRINT "SORRY, TRY AGAIN"
310 GOTO 140
```

הדיאגרמה הבאה מבהירה את מבנה התכנית :





אפשר לשנות תכנית זו עבור כפל, חיסור או חילוק. אבל, כשמשנים עבור חילוק וחיסור ורוצים שהתשובות תהיינה מספרים שלמים חיוביים, צריך להיות זהירים לגבי הוראות ההדפסה למחשב.

התכנית הבאה של תרגול פועלת בדומה לקלפי משחק המתרגלים חשבון פשוט. השחקן בוחר את המספרים שמהם הוא רוצה לתרגל כפל, למשל, והמחשב מציג תרגילים ובודק את התשובות. בשינויים קלים, יכולה התכנית הזו לשמש לחיבור, חיסור או חילוק. כמו קודם, יש לנו בעיה בתרגילי חילוק אם אנו רוצים להגדיל את עצמנו לתוצאות שהן מספרים שלמים. הפתרון לבעיה זו מוצג שוב אחרי ההדפס (LIST) של התכנית הבאה.

```
10 REM *   SIMPLE VERSION OF TIMES
20 REM *           TABLE DRILL
30 REM *  (C) 1981  BY HERBERT KOHL
40 REM *
50 DIM NAME$(20)
60 PRINT "[CLEAR]WHAT IS YOUR NAME";
E";
70 INPUT NAME$
80 PRINT "[CLEAR]WHICH NUMBER FROM YOUR
TIMES TABLE"
90 PRINT "DO YOU WANT TO PRACTICE, ";NAM
E$;
100 INPUT N
120 REM * SIZE OF NUMBERS DEPENDS ON
125 REM * VALUE OF X IN LINE 130
130 LET X=INT(RND(1)*10)+1
140 PRINT "WHAT IS THE ANSWER, ";NAME$
150 PRINT X;" * ";N;" = ";
160 INPUT Z
170 IF Z=X*N THEN GOTO 200
180 PRINT :PRINT "TRY AGAIN...":PRINT
190 GOTO 150
200 PRINT :PRINT "YOU GOT IT!"
210 PRINT "HERE'S ANOTHER ONE TO TRY"
220 PRINT
230 GOTO 130
```

בנית תכנית לתרגול חילוק הוא כבר אתגר קצת יותר גדול. סיבה אחת היא, שחילוק ב-0 הוא חסר משמעות בחשבון רגיל. שנית, מחשב אטארי בנוי לתת תשובות לחישובים מתמטיים עד למספר כזה של מקומות אחרי הנקודה העשרונית, שרוב הילדים ורוב המבוגרים אינם זקוקים לו ואינם מעונינים בו.



הצגנו כבר תכנית שמספקת תרגול פשוט בחילוק. יש אתגר מענין בשינוי התכנית כך שתוכל לטפל בכל בעית חילוק, תבדוק כדי לוודא שלא יתקיים חילוק ב-0, ותרשה לך לבחור לבד את ערכו של המחלק. ויש לנו כל אוצר המלים של BASIC כדי לבצע זאת !

מחשב אטארי יכול לבצע חישובים מתמטיים מסובכים, ומסיבה זו אפשר להשתמש בו לבנית תכניות תרגול ללימוד אלגברה. המחשב יכול לבנות משוואות, לפתור אותן, לשאול אותך על הפתרונות ולתקן את תשובותיך. הנה תכנית עם משוואות חיבור פשוטות שהתשובות להן הם מספרים שלמים חיובים. זו יכולה להיות דרך טובה לפגישה ראשונה עם האלגברה.

```
10 REM *  SOME ALGEBRA WITH POSITIVE
20 REM *      INTEGER SOLUTIONS
30 REM *  equations in one variable
40 REM *  (C) 1981 BY HERBERT KOHL
50 PRINT "[CLEAR]"
100 LET X=INT(RND(1)*11)
110 LET Y=INT(RND(1)*11)
120 IF X>Y THEN GOTO 100
130 PRINT "HERE IS A SIMPLE ALGEBRA PROB
LEM."
140 PRINT
150 PRINT "WHAT IS A CORRECT VALUE FOR W
":PRINT "IF ";
160 PRINT X;" + W = ";Y
170 PRINT "W=";:INPUT W
180 IF W=Y-X THEN GOSUB 200:GOTO 100
190 PRINT "SORRY... TRY AGAIN":GOTO 150
200 REM *
210 REM * GOOD ANSWER
220 REM *
230 PRINT :PRINT "WELL DONE!  NOW TRY TH
IS ONE"
240 RETURN
```

התכנית הבאה משתמשת במשוואות מן הסוג : W=AX+AY

```
10 REM * SETTING UP ANOTHER SIMPLE
20 REM *     DRILL IN ALGEBRA
30 REM * (C) 1981 BY HERBERT KOHL
40 REM *
100 LET X=INT(RND(1)*21)+1
110 LET Y=INT(RND(1)*21)+1
120 PRINT "[CLEAR]"
130 PRINT "HERE IS A SIMPLE EQUATION."
135 PRINT "PICK AN A AND SOLVE IT FOR W"
140 PRINT "(A*";X;") + (A*";Y;") = W"
150 PRINT "CHOOSE AN A";
160 INPUT A
170 LET W=A*X+A*Y
180 PRINT :PRINT "WHAT IS YOUR SOLUTION ";
190 INPUT N
200 IF N=W THEN GOTO 300
210 PRINT "TRY AGAIN...":GOTO 190
300 PRINT :PRINT "YOU GOT IT!"
999 END
```



כדאי לשים לב למילה END בשורה 999.
הפקודה הזו מבצעת בדיוק את מה שהיא אומרת.
היא מסיימת את התכנית באופן רשמי. במקרה זה, אחרי שנתת תשובה נכונה אין המחשב מספק משוואה חדשה. על מנת להתחיל את התכנית מחדש, צריך לתקתק RUN וללחוץ על מקש RETURN. לעתים רוצים שהמחשב יריץ תכנית ואז יפסיק. זה תפקידה של הפקודה END.



הנה עוד תכנית שמספקת תרגול של כמה מושגים אלגבראיים ומתרגלת "יכולת חשיבה אינדוקטיבית". אין הכרח להגביל תרגול מתמטי לפעולות החשבון היסודיות בלבד. ללמוד לחשוב כהלכה חיוני לפתרון בעיות יותר מורכבות, ו"מיומנויות חשיבה" דורשות גם הן תרגול !
מטרת המשחק הבא היא לקבוע מספר שהוא הבא בתור בתוך סדרה של מספרים. המשחק מומלץ לתלמידים בכתות העליונות של בית הספר היסודי, אבל גם לתלמידים בכתות גבוהות יותר ולמבוגרים. הענין כרוך באוצר מלים גדול יותר ב- BASIC משידוע לך עד כה.
יחד עם זאת, כדאי לתקתק את התכנית לתוך המחשב, להריץ אותה ולנסות לערוך כמה שנויים.

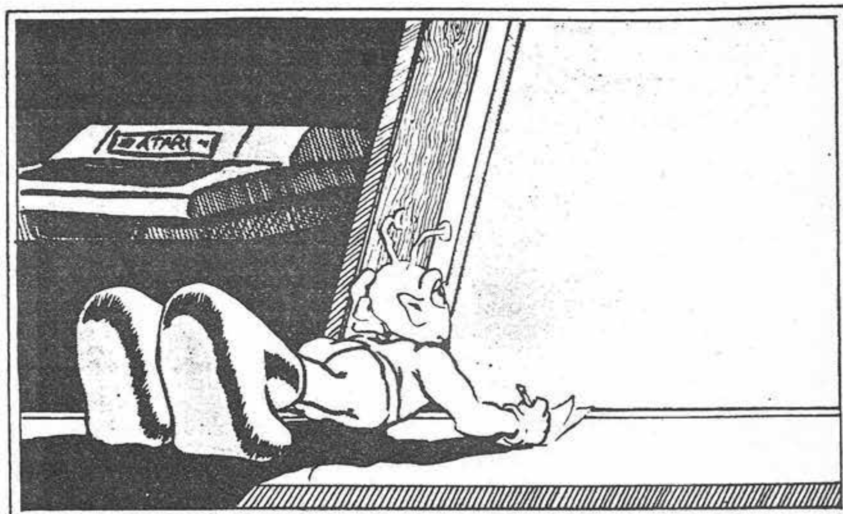
כשתריץ את התכנית, תראה שהיא קודם מבקשת משחקן שיפעל כ"קובע החוקים" לתקתק שלשה מספרים. הדבר נעשה כדי ששני שחקנים יוכלו לשחק : "קובע החוקים" יתקתק את המספרים , ואז מנסה השחקן השני לנחש מה החוק ע"י התבוננות בכל מספר חדש בסדרה. "קובע החוקים" יכול להיות הורה, מורה או חבר. המשחק הוא תרגול טוב במיוחד לאלגברה מכיון שהערכים שבשימוש בתכנית הם כולם בעלי אותה צורה (פונקציה ליניארית של הנוסחה $Y=A*X+B$).

נסה את המשחק וראה אם אפשר לשפר בו משהו !

```
10 REM *       <<< WHAT NEXT >>>
20 REM *  (C) 1981  by Ted M. Kahn
30 REM *   -A game of practice in
40 REM *  inductive thinking skill
50 GOSUB 300
60 GRAPHICS 0
70 DIM A$(1),GUESS$(7)
80 REM * MYSTERY NUMBER = A*X+B
90 GRAPHICS 0:PRINT "FOR THE RULE-MAKER
ONLY:":PRINT
100 TRAP 100:PRINT "INPUT A VALUE FOR X
(1-10)";:INPUT X:IF X<1 OR X>10 THEN X=A
BS(INT(X/10))
110 TRAP 110:PRINT :PRINT "INPUT A VALUE
 FOR A (0-10";:INPUT A:IF A<0 OR A>10 TH
EN A=ABS(INT(A/10))
120 TRAP 120:PRINT :PRINT "INPUT A VALUE
 FOR B (0-10)";:INPUT B:IF B<0 OR B>10 T
HEN B=ABS(INT(B/10))
130 TRAP 130:PRINT "HOW MANY GUESSES ALL
OWED";:INPUT GUESS
140 LET GUESS=GUESS+1:LET GUESS$="GUESSE
S"
150 GRAPHICS 2:REM * BIG LETTERS
160 COLOR 1
170 LET GUESS=GUESS-1:IF GUESS<1 THEN GO
TO 260
180 IF GUESS=1 THEN LET GUESS$="GUESS"
190 PRINT "[CLEAR]WHAT'S THE NEXT NUMBER
 IN THIS SERIES?"
200 PRINT "(YOU HAVE ";GUESS;" ";GUESS$;
" LEFT)";
210 PRINT #6;" ";X;
```

```
220 LET X=A*X+B
230 TRAP 230:INPUT Y
240 IF Y=X THEN GOSUB 440:GOSUB 610:GOTO
 90
250 GOSUB 550:GOTO 170
260 PRINT "[CLEAR]THE NEXT NUMBER WAS ";
X
270 PRINT "THE SECRET RULE WAS ";A;" * X
 + ";B
280 GOSUB 610
290 END
300 REM *
310 REM * TITLE PAGE
320 REM *
330 GRAPHICS 1
340 COLOR 1
350 POSITION 5,4
360 PRINT #6;"what next?":PRINT #6
370 PRINT #6;"  MYSTERY NUMBERS"
380 POSITION 4,10
390 PRINT #6;"(C) 1981 BY"
400 PRINT #6;"     ted m. kahn"
410 PRINT "(PRESS START TO BEGIN...)"
420 IF PEEK(53279)<>6 THEN 420
430 RETURN
440 REM *
450 REM * CORRECT ANSWER
460 REM *
470 GOSUB 680
480 FOR I=1 TO 5
490 GRAPHICS 2:COLOR 2
500 POSITION 4,5:PRINT #6;"you got it!"
510 FOR J=50 TO 100:SOUND 1,J,10,10:NEXT
 J
520 SOUND 1,0,0,0
530 NEXT I
540 RETURN
550 REM *
560 REM * INCORRECT ANSWER
570 REM *
580 PRINT :PRINT "NO, THAT'S NOT IT.":PR
INT "HERE'S THE NEXT ONE..."
590 GOSUB 680
600 RETURN
```

```
610 REM *
620 REM * WANT ANOTHER GAME?
630 REM *
640 PRINT "DO YOU WANT TO PLAY AGAIN";
650 INPUT A$
660 IF A$="Y" THEN RETURN
670 GOTO 290:REM * END OF GAME
680 REM *
690 REM * PAUSE (WAIT) ROUTINE
700 REM *
710 FOR WAIT=1 TO 300:NEXT WAIT
720 RETURN
```

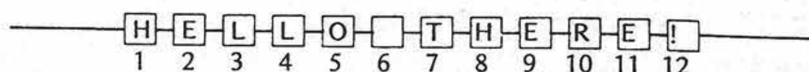


קצת דקדוק אנגלי וכתיב: השימוש ב"משתני מחרוזת"

עלינו להציג כמה מלים חדשות ב-BASIC על מנת לכתוב תכניות תרגול או משחקים עם אותיות ומלים. הסיבה לכך היא שניב BASIC של אטארי מטפל במלים באופן שונה מזה שהוא מטפל במספרים. כדי להבין זאת, צריך קודם להבין את המושג מחרוזת תוים. אמנם השתמשת במחרוזות תוים לפני כן, אבל לא ידעת כל כך. בכל פעם שנתת פקודת PRINT, בעצם הדפסת מחרוזת תוים. למשל :

```
PRINT "HELLO THERE!"
```

במשפט הזה, האותיות, הרווחים וכן סימן הקריאה (!) הם כולם תוים והמשפט כולו הוא מחרוזת תוים. אפשר לחשוב על כך כעל מחרוזת רגילה של חרוזים, כאשר כל חרוז מכיל תו אחד. בדוגמה למעלה מכילה המחרוזת 12 תוים בדיוק, כפי שנראה מיד :





כל דבר שמופיע בין מרכאות ב BASIC של אטארי הוא מחרוזת תוים. משתמשים בה כלשונה, ממש כפי שרואים אותה. כמעט כל דבר יכול להיות חלק ממחרוזת תוים, כולל מספרים, סימנים גרפיים מיוחדים (אלה המופקים ע"י לחיצה על (CTRL) ובאותו זמן תקתוק על מקש אחר), חיצי-הכוון של הנתב (למעלה, למטה, ימינה ושמאלה). אבל, במקרה הזה צריך ללחוץ על (ESC) לפני הלחיצה על החיצים. לכל מחרוזת יש אורך, שאותו מודדים באמצעות המספר המדויק של תוים במחרוזת; ואפשר להדפיס כל מחרוזת שהיא על המסך.

(4 תוים; מחרוזת תוים עם מלה אחת) "NOUN" ◦

(17 תוים) " 12 LEAVE ME ALONE" •

(18 תוים) " this is lower case" •

(13 תוים; החלפת כתב ורקע " inverse video" • מתקבלת ע"י לחיצה על המקש עם סמל אטארי אחרי המרכאות הראשונות ולפני האחרונות).

(ארבעה סמנים גרפיים המתקבלים ע"י " ●◢◢● ". (CTRL) עם מקשי T ו- H ; ● הוא (T) ו- ◢ הוא (H).

מחרוזות, כמו מספרים, ניתנות לביצוע מידי או יכולות להיות מאוחסנות בזכרון באמצעות שם של משתנה. השתמשנו כבר בשמות משתנים רבים עבור מספרים, כמו X, Y או NUMBER. אם נרצה לאחסן מחרוזות תוים בזכרון, עלינו לעשות את הדברים הבאים :

1. לשמור מקום למשתנה המחרוזת באמצעות הכרזה הנקראת DIM (מן המלה האנגלית Dimension). הכרזת ה- DIM עבור מחרוזת אחת או יותר מורכבת מן המילה DIM , אחריה השם של משתנה המחרוזת שבסופו תמיד מופיע סימן הדולר ($), והמספר המירבי של תוים שעל המשתנה לאחסן.

2. כדי לתת ערך למשתנה, נשתמש בבטויים LET או INPUT.

3. ברגע שיש ערך למשתנה המחרוזת (כל מחרוזת שהיא-מלים, אותיות, סימנים וכו' שתכניס לתוכה), אפשר להדפיס את הערך הזה, באותו אופן שאפשר להדפיס את הערך של משתנים מספריים כמו X או Y .

האם זה נשמע מבלבל ? לא נורא ! עם קצת סבלנות ותרגול של כמה דוגמאות פשוטות, הרעיון יתברר במהירות.



נניח שברצוננו לשמור מקום בזכרון המחשב למלה בת 10 אותיות או פחות. נתחיל במתן שם למשתנה מחרוזת שיכיל את המילה. בסוף שם המשתנה יבוא סימן דולר ($) כדי לציין שהמשתנה יהיה תמיד עבור מחרוזות תוים. שם המשתנה יכול להיות A$,B$, NAME$ או W10$. למעשה, יכול שם המשתנה להתחיל בכל אות שהיא, כשאחריה צירוף של עד 254 אותיות או מספרים ! אף פעם אין להשתמש ברווח כחלק משם המשתנה, אך רווחים יכולים להיות חלק מן המחרוזת עצמה. אנו ממליצים לבחור שמות משתנים שהם בעלי משמעות עבורך, כל עוד אינם מכילים מלים שמורות של ה BASIC של אטארי. △

בואו נקל על החיים ונבחר בשם A$. כדי לומר למחשב לשמור מקום בזכרון למילה או תוים ב A$, נשתמש בהכרזה DIM . אחרי המילים DIM A$ נשים את המספר הגבוה ביותר של תוים שנרצה לאחסן במשתנה A$, ונציין אותו בתוך סוגריים. וכך זה נראה :

```
10 DIM A$(50)
```

דבר זה אומר למחשב שהמשתנה בשם A$ יוכל להכיל אותיות או מחרוזות עד לאורך של 50 תוים.

כעת משמרנו את המקום, אפשר לתקתק כל מלה או מחרוזת עד לאורך של 50 תוים ולאחסן אותה בזכרון תוך כדי שימוש ב INPUT . נוסיף את השורה הזאת לתכנית :

```
20 INPUT A$
```

△ השמוש במלים כמו NEXT$, LET$, RUN$, GOTO$ וכו' עלול לגרום למחשב בלבול בין שמות המשתנים לבין הפקודות של BASIC.



אם רוצים לראות על המסך את מה שהוכנס באמצעות INPUT, נוסיף את השורה הבאה :

```
30 PRINT A$
```

עכשו, הבה נוסיף לתכנית הודעה שתאמר לשחקן מה לעשות. נוסיף את השורה הזאת - והחופש בידך לכתוב את המילים שלך בין המרכאות :

```
15 PRINT "WHAT'S YOUR WORD";
```

פקודת LIST תראה את כל התכנית על המסך. מה לדעתך תעשה התכנית ? נריץ (RUN) ונראה ! מה קורה אם מכניסים מילה או משפט בני יותר מ-50 תוים ? כדאי לנסות קצת את העניין של שמירת מקום בזכרון באמצעות הכרזת DIM ע"י שנוי המספר שבסוגריים בשורה 10.

כפי שכבר ודאי גלית, התכנית תקצץ את כל התוים שמעבר למספר התוים שצוינו ב DIM עבור A$. אבל זה דוקא די שמושי. נשנה קצת את התכנית ונריץ :

```
10 DIM A$(1)
```

ונוסיף :

```
40 GOTO 15
```

ננסה כמה מחרוזות שונות. (למשל YES, NO, ADDITION וכו') מה מאוחסן בתוך A$? התכנית הקטנה הזאת מאפשרת לנו לבדוק תשובות פשוטות של כן-לא.מאחר ש A$ יכיל רק את האות הראשונה של מה שתוקתק כתגובה ל INPUT, נוכל לדעת מיד איזו מילה נכתבה.

A$ יכיל Y אם תקתקת YES, ויכיל N אם תקתקת NO.
אפשר להשתמש בתכנית גם כדי לבקש מן השחקנים לבחור מרשימה, או תפריט, של אפשרויות, כמו לבחור בסוג תרגילי החשבון שברצונם לתרגל : A עבור חיבור (מן המילה האנגלית ADDITION ,חיבור) S עבור חיסור (SUBTRACTION)וכו'. דוגמה לתכנית מסוג זה נמצאת בסוף הפרק.

השתמשנו כבר ב DIM ו- INPUT עם NAME$ בשלב מוקדם יותר בפרק, כדי לתת גוון אישי לתכנית. הסתכל בהתאמה הזו של תכנית קודמת של תרגול. עכשו צריך להיות ברור כיצד משתמשים ב NAME$

```
10 DIM NAME$(20)
20 PRINT "WHAT IS YOUR NAME";
30 INPUT NAME$
40 PRINT "HI, ";NAME$
45 PRINT "HERE'S AN ADDITION PROBLEM FOR
 YOU TO TRY"
50 PRINT "2+2=";
60 INPUT X
70 IF X=4 THEN PRINT "GOOD JOB ";NAME$
80 IF X<>4 THEN PRINT "BETTER LUCK NEXT
TIME, ";NAME$
```

ראוי לשים לב שאין מרכאות ('') מסביב ל NAME$.
אילו היו מרכאות מסביב ל NAME$, בשורות 70,40 או 80
היה המחשב מדפיס את המילה NAME$ ולא את ערך המשתנה,
כלומר את המלה המאוחסנת במשתנה הנקרא NAME$.



לפני שנציג כמה תרגילים ומשחקים המשתמשים באותיות
ובמלים, עלינו להציג מרכיב נוסף של משתני מחרוזת.
כפי שאמרנו קודם, מורכבות המחרוזות מצירוף של תוים
בודדים. כל מחרוזת המאוחסנת במשתנה מחרוזת היא
בעלת אורך מסוים ובעלת מספר מסוים של תוים בודדים
בכל עת. נוכל לגלות מה מכיל כל תו בודד ע"י השמוש
בפונקצית המחרוזת A$(I,J). למשל, הנה תכנית פשוטה
ביותר שמטרתה לתרגל ילדים בזהוי מלים המתחילות
באותיות שונות של האלפבית. המחשב בוחר אות באופן
אקראי ומבקש את הילד לתקתק מילה שמתחילה באות הזו.
אחרי תגובת הילד, אומר לו המחשב אם בחר במילה
שמקיימת את התנאי המקורי.



```
10 REM * LETTER IDENTIFICATION GAME
20 REM *     FOR YOUNG CHILDREN
30 REM * (C) 1981 BY FRONA STAR &
40 REM *       TED M. KAHN
50 DIM A$(1),ALPHABET$(26),X$(1),WORD$(2
0)
60 LET ALPHABET$="ABCDEFGHIJKLMNOPQRSTUV
WXYZ"
70 GRAPHICS 2
80 PRINT "[CLEAR]LET'S PLAY A LETTER GAM
E."
85 PRINT "I'LL TYPE A LETTER AND YOU TYP
E A ":PRINT "WORD THAT BEGINS WITH MY LE
TTER":GOSUB 500
```

```
90 PRINT "[CLEAR]"
95 PRINT #6;"[CLEAR]"
100 LET L=INT(RND(1)*26)+1
110 A$=ALPHABET$(L,L)
120 POSITION 9,4
130 PRINT #6;A$
140 INPUT WORD$:LET X$=WORD$(1,1)
150 IF X$=A$ THEN GOSUB 200:GOTO 90
160 PRINT "NO, ";WORD$;" DOESN'T BEGIN W
ITH ";A$;".":PRINT "TRY ANOTHER WORD"
170 GOTO 140
200 REM *  REWARD FOR CORRECT WORD
210 PRINT #6;"[CLEAR]"
220 POSITION 1,4
230 PRINT #6;A$;" IS FOR ";WORD$
240 GOSUB 500
250 RETURN
500 REM * PAUSE/DELAY
510 FOR WAIT=1 TO 1000:NEXT WAIT
520 RETURN
```

תכנית זו משתמשת בכמה מחרוזות תוים, שעל כולן הוכרז ע"י DIM בשורה 50. המחרוזת ALPHABET$ מכילה את 26 האותיות של האלפבית האנגלי, לפי הסדר (שורה 60). WORD$ משמש לאחסון מלה שיתקתק הילד (שורה 140). בשורה 100 נבחר מספר אקראי בין 1 ו-26, ואז משתמשים באות הנמצאת במקום זה כגרוי לתגובת הילד (שורה 110).

נסתכל בשורה זו ובשורה 140 ביתר תשומת לב :

110 A$=ALPHABET$(L, L)

זהו קיצור של : LET A$=ALPHABET$(L, L)



שורה זו אומרת לקחת את הערך של L ולהשתמש בה כדי למצוא את הערך של "תת המחרוזת" ALPHABET$ שמתחילה במספר התו L ומסתימת במספר התו L. מאחר שתת מחרוזת זו מתחילה ומסתימת באותו תו, יבחר התו הנמצא במקום ה L באלפבית. למשל, אם ערכו של L היה נבחר להיות 5, האות החמישית , E היתה נבחרת מתוך ALPHABET$ ומועברת כערכו של A$.

באופן דומה,

140 INPUT WORDS$: X$=WORD$(1, 1)

אומרת לתקתק מלה ולהעביר את התו (האות) הראשון של המילה למשתנה X$.

זוהי גירסה פשוטה מאד של המשחק, אך יש בה פרס נחמד של הדפסת משפט הבנוי על המילה שבוחר הילד, אם היא מתחילה באות שבחר המחשב : ההדפסה מופיעה באותיות גדולות ע"י GRAPHICS2. התכנית איננה בודקת אם הכתיב נכון. תרגול בכתיב אנגלי יבוא עוד מעט.

תרגיל כתיב

הנה תכנית שאפשר להשתמש בה כדי לתרגל רשימה של חמש מילים מתוך רשימה של מלים להכתבה.
המילים יכולות להיות גם בשפות אחרות, כמו צרפתית או ספרדית. למרות שהתכנית מוגבלת ל-5 מלים בלבד, כדאי לאחסן אותה על סרט או על תקליטון, משום שאפשר יהיה להשתמש בה כל פעם עבור רשימת מילים אחרת (שתוכנס ע"י התלמיד).

בחלקה הגדול משתמשת התכנית במילים ב BASIC שכבר מוכרות לך. בשורה 125 יש מילה אחת חדשה, ON-GOSUB :

```
125 ON W GOSUB 200, 300, 400, 500, 199
```

המשפט החשוב הזה מאפשר לנו לציין מספרים שונים של קריאה לתת-תכניות, תלוי בערך של המשתנה המובא אחרי המלה ON (במקרה זה, שם המשתנה הוא W). משפט יחיד זה בא במקום חמשת משפטי BASIC הרשומים מטה :

```
125 IF W=1 THEN GOSUB 200
126 IF W=2 THEN GOSUB 300
127 IF W=3 THEN GOSUB 400
128 IF W=4 THEN GOSUB 500
129 IF W=5 THEN GOSUB 199
```



המשפט הזה תמיד מניח שלמשתנה (כאן W) יש ערך במספרים שלמים, ומספרי השורות עבור כל ה GOSUB צריכים להיות מצוינים במדויק ובהתאמה לערכים 1,2,3,4 של W (לפי הסדר). בתכנית זו תת התכניות בשורות 200-500 הן כולן בנות שורה אחת בלבד.

בתכנית הזו, אורך המילים הוא לא יותר מ-10 אותיות. אפשר להגדיל את אורכן ע"י שנוי הכרזות ה DIM בשורות 20-27

```
10 REM * SIMPLE SPELLING DRILL PROGRAM
15 REM *   (C) 1980 BY HERBERT KOHL
20 DIM A$(10),B$(10),C$(10),D$(10)
25 DIM E$(10),M$(10)
30 PRINT "[CLEAR]"
35 PRINT "WRITE DOWN 5 OF YOUR ";
40 PRINT "SPELLING WORDS":PRINT "(ONE PE
R LINE, FOLLOWED BY RETURN)"
45 INPUT A$,B$,C$,D$,E$
50 PRINT "[CLEAR]"
55 PRINT "ONE OF YOUR SPELLING WORDS WIL
L BE"
60 PRINT "FLASHED ON THE SCREEN BRIEFLY.
":PRINT "STUDY IT."
65 GOSUB 700
70 GRAPHICS 2
75 FOR W=1 TO 5
80 PRINT #6;"[CLEAR]"
85 POSITION 6,4
90 PRINT #6;A$
95 GOSUB 700:PRINT #6;"[CLEAR]"
100 PRINT "NOW TRY TO SPELL THE WORD"
105 INPUT M$
110 IF M$<>A$ THEN GOSUB 600:GOTO 105
115 POSITION 4,4:PRINT #6;"WAY TO GO!"
120 GOSUB 700
125 ON W GOSUB 200,300,400,500,199
130 NEXT W
199 END
200 A$=B$:RETURN
300 A$=C$:RETURN
400 A$=D$:RETURN
500 A$=E$:RETURN
600 REM * INCORRECT SPELLING
610 PRINT "NOT QUITE...  TRY AGAIN"
620 RETURN
700 REM * PAUSE ROUTINE
710 FOR WAIT=1 TO 1000:NEXT WAIT
720 RETURN
```



גירסה שניה של אותה התכנית היא מעט גמישה יותר, ומשתמשת בשתי מלים חדשות ב BASIC : READ ו- DATA. המלים האלה, כמו FOR ו- NEXT או GOSUB ו-RETURN, מופיעות כזוג: READ A$ מבצעת אותה הפעולה כמו INPUT A$,

אלא שהמחשב קורא (READ) את התוכן של A$ מתוך המשפט הראשון של DATA שנקרה בדרכו.(DATA מן המילה האנגלית שפירושה "נתונים"). משפט DATA יכול להופיע בכל מקום בתכנית ומכיל את נתוני המחרוזת(או נתונים מספריים, אם זה עבור משתנים מספריים) שצריך לקרוא אותם לתוך המשתנה A$ במשפט READ . "לקרוא משהו לתוך משתנה" פירושו לקרוא את הנתון המסוים, מספרי או מחרוזת, ולהציב את הנתון הזה כערך של המשתנה. בכל פעם שמבצעים ביטוי READ , "נקרא" הפריט הבא ברשימת ה-DATA. פירוש הדבר יכול להיות גם ללכת למשפט חדש של DATA (אם הוא ישנו) כדי למצוא את הפריט הבא ברשימה.

בתכנית שנביא כעת, הכנסנו את נתוני המילים להכתבה בשורה 600 והלאה. כרגע יש בהן מילים בצרפתית, המייצגות רשימה או אוצר מלים. כמובן שאפשר להחליף אותן בכל חמש מלים אחרות בעלות אורך של עד 10 אותיות . אם ברצוננו להגדיל את מספר המלים לתרגול, פשוט נשנה את שורה 100 (התחלת הלולאה FOR-NEXT), כך ש- W ילך מ-1 ועד למספר הכולל של מילים שברשימת ה- DATA ; נוסף לכך, אפשר להוסיף מילים ככל שנחוץ במשפטי DATA עוקבים, אחרי שורה 600. צריך לזכור להפריד בין המילים ע"י פסיק (,).

ושוב, אפשר לאחסן את התכנית לשמוש מאוחר יותר, בכל זמן שמישהו יצטרך ללמוד רשימה של מלים להכתבה.

```
10 REM *  << SPELLING DRILL #2 >>
20 REM *      copyright 1980 by
30 REM *  HERBERT KOHL & TED KAHN
40 DIM A$(10),X$(10)
50 PRINT "[CLEAR]EACH WORD WILL BE "
60 PRINT "BRIEFLY FLASHED ON THE SCREEN"
70 PRINT "HERE THEY ARE..."
90 GOSUB 500
100 GRAPHICS 2
110 FOR W=1 TO 5
120 READ A$
130 PRINT "NOW STUDY THE NEXT WORD"
140 PRINT #6;"[CLEAR]"
150 POSITION 0,4:PRINT #6,A$
160 GOSUB 500
170 PRINT #6;"[CLEAR]"
180 PRINT "[CLEAR]NOW SPELL THE WORD"
190 INPUT X$
```

```
200 IF X$=A$ THEN GOSUB 300:GOTO 220
210 IF X$<>A$ THEN GOSUB 400:GOTO 140
220 NEXT W
230 END
300 REM * CORRECT SPELLING
310 PRINT "YOU GOT IT!  GOOD GOING!"
320 GOSUB 500
330 RETURN
400 REM * INCORRECT SPELLING
410 PRINT "NOT QUITE... TRY AGAIN."
420 GOSUB 500
430 RETURN
500 REM * PAUSE/DELAY
510 FOR WAIT=1 TO 1000:NEXT WAIT
520 RETURN
599 REM * DATA IS FROM LINE 600 ON-
600 DATA MANGER,PORTER,VENIR,FAIRE,TOUCH
ER
```



מושגים

☆ תת-תכנית- תת תכנית היא תכנית בתוך תכנית. היא יכולה להיות עיטור, סוף לתכנית, הנחיה לשחקן לנסות שנית. תת תכניות הן דרך נוחה לקבץ חלקים שונים של תכנית. למשל, זוהי תת תכנית פשוטה של השהיה :

```
1000 REM PAUSE ROUTINE
1110 FOR WAIT=1 TO 1000:NEXT WAIT
1199 RETURN
```

מילים ב- BASIC

☆ GOSUB ו- RETURN- GOSUB משמשת כדי לומר למחשב "לקפוץ" לתת תכנית שמתחילה במספר השורה המצויין ; RETURN מסיימת את תת-התכנית ושולחת את התכנית חזרה למקום שבו הפסיקה. העצמה של תת התכניות היא שאפשר להשתמש בהן לעתים קרובות בהרבה תכניות שונות כדי להשיג אותה מטרה. הנה שמוש בתת תכנית להשהייה :

```
10 PRINT "GUESS A NUMBER FROM 1 TO 10"
20 INPUT X
30 GOSUB 1000
40 IF X=7 THEN PRINT "YOU GOT IT"
:    etc.
:
```

```
1000 FOR Z=1 TO 1,000
1010 NEXT Z
1020 RETURN
```

כמובן, התכנית אינה גמורה, אבל היא מדגימה את השמוש בתת תכניות. בשורה 30 קופצת התכנית לשורה 1000, אשר גורמת להשהיה במשחק. אחרי ההפסקה, מורה שורה 1020 למחשב לחזור לשורה 40 ולהמשיך עם שאר התכנית.

☆ ON W GOSUB – הביטוי ON-GOSUB מאפשר למחשב לעבור דרך תת תכניות שונות בהתאם לערך של המשתנה W.
הערך של W חייב להיות מצויין בבהירות, למשל ע"י שמוש בלולאה פשוטה של FOR-NEXT, כמו זו :

```
100 FOR W=1 TO 3
110 PRINT "HERE'S SOMETHING INTERESTING TO READ"
120 ON W GOSUB 200, 300, 400
130 FOR Y=1 TO 250:NEXT Y
140 NEXT W
150 END
200 PRINT "IT WAS THE BEST OF TIMES"
210 PRINT "IT WAS THE WORST OF TIMES"
220 RETURN
300 PRINT "E=(M*C*C)"
310 RETURN
400 PRINT "WHAT IS BLACK AND WHITE AND PURPLE ALL
    OVER"
410 RETURN
```

☆ END – משמעותו כפשוטו, סיים (END) את התכנית.

☆ DIM A$(X) – הביטוי DIM שומר מספר קבוע של תאים בזכרון המחשב עבור המשתנה A$.
המקומות האלה יכולים להכיל אותיות ותוים מיוחדים וכן מספרים.
DIM A$ (10) שומר 10 מקומות עבור A$.
פקודת INPUT משמשת לאחסון ב- A$, ואפשר להדפיס את תכנו של A$ מתי שרוצים. הנה דוגמה :

```
10 DIM A$(25)
20 PRINT "WHAT IS YOUR NAME?"
30 INPUT A$
40 PRINT "HELLO■";A$
```

A$(X,Y) – A$(X,Y) מאפשר להשיג אות מסוימת או קבוצת אותיות מתוך מחרוזת שעליה מכריזים ע"י DIM .
אם A$=ABCDE, אז

A$(1,3) = A,B,C שלש האותיות הראשונות

A$(4,5) = E,D. האותיות הרביעית והחמישית

A$(1,1) = A ובאופן כללי, האות הראשונה

A$(X,X) יתן מה שנמצא במקום ה- X של A$

☆ READ-DATA – מאפשרת להכניס נתונים רבים למשתנה.
למשל, הנה דוגמת תכנית READ-DATA

```
10 FOR X=1 TO 3
20 READ X
30 PRINT X+40
40 NEXT X
50 DATA 17, 18, 19
```

התכנית הזו אומרת, שעבור X שוה 1 עד 3, על התכנית לקרוא (READ) את הנתונים (DATA) בשורה 50, להוסיף 40 לכל X ולהדפיס את התשובה. התוצאה של תכנית פשוטה זו תהיה :

```
57
58
59
```

תכנית כמו זו שלפנינו תאפשר להכניס 100 מספרים שונים בשורה 50, להעלות כל מספר ברבוע ולהדפיס את התוצאה :

```
10 FOR X=1 TO 100
20 READ X
30 PRINT X*X
40 NEXT X
50 DATA
```

ואלה המלים ב- BASIC שלמדנו בחלק הראשון :

A$	IF . . . THEN . . .	
	INPUT A$	RND(1)
A$(X, Y)	INPUT X	RUN
	INT(X)	SETCOLOR

CLOAD	INT(RND(1))	SETCOLOR X, Y, Z
	INT(RND(1)*10)+1	SOUND W, X, Y, Z
COLOR X	LET X=	
CSAVE	LIST	X<Y
CONT	NEW	X<=Y
		X>Y
	ON W GOSUB	X>=Y
DIM A$(X)	PLOT X, Y	X< >Y
DRAWTO X, Y	POSITION X, Y	X+Y
END	PRINT	X-Y
FOR-NEXT	PRINT #6	X*Y
FOR-STEP-NEXT		X/Y
GOSUB-RETURN	READ-DATA	X∧Y
GOTO	READY	< >
GRAPHICS 2		
GRAPHICS 7	REM	

מושגים

הסתעפות	לולאה
לולאות השהייה	לולאות מכוננות
הודעות שגיאה	תכנית
מצבים גרפיים	תת תכנית
מספר שורה	חלון הטקסט
שורה לוגית	

סימני פיסוק

" "
;
( )
:
,

מקשים מיוחדים

[BACKSPACE]	[INSERT]	[SYSTEM RESET]
[BREAK]	[RETURN]	[CTRL] [INSERT]
[CTRL]	[SHIFT]	[CTRL] [DELETE]
[DELETE]	[SHIFT] [CLEAR] or [CTRL] [CLEAR]	[CTRL] [↑] (up)
[ESC]	[SHIFT] [DELETE]	[CTRL] [↓] (down)
[ESC] [CLEAR]	[SHIFT] [INSERT]	[CTRL] [←] (left)
[ESC] [CTRL]		[CTRL] [→] (right)

חלק שני

משחקים
למחשב האישי
של אטארי

בחלק הראשון של הספר השתמשנו במשחקים כדי להציג מבוא ל- BASIC של אטארי. בחלק זה נביא משחקים רבים לשמוש במחשב אטארי, 400 או 800 : אם כפי שהם מופיעים כאן, או עם תוספת עיטור אישי שלך. כמה מתכניות המשחקים הפשוטות יותר תראינה לך קלות להבנה, אם למדת ברצינות את החלק הראשון. אולם, רבות מן התכניות כאן הן מסובכות למדי ודורשות ידיעה עמוקה יותר בתכנות ממה שאנחנו יכולים להביא בספר זה. אבל זו לא סיבה לותר על התכניות האלה. אפשר לתקתק אותן לתוך הזכרון של המחשב, ולשחק גם כך. כדי להמנע מעבודה מיותרת, כדאי לשמור אותן על קסטה או תקליטון. ניתן גם לראות את התכניות על המסך (באמצעות LIST) ולשנות בהן מדי פעם. יהיה לך קל לזהות תת-תכניות שהזכרנו קודם לכן, משפטי PRINT, לולאות והסתעפויות. אם תהיה לך בעיה כלשהי- המדריך למשתמש ל- BASIC של אטארי עומד לרשותך בכל עת, ליעוץ בענין פקודות BASIC שאינן מובנות לך. על כל פנים, הידע בתכנות יגבר ככל שהשמוש במחשב יהיה תכוף יותר.

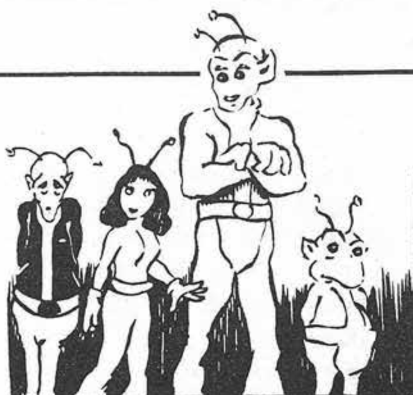
פרק ד'



משחקי הגיון ומספרים

המשחקים הפשוטים ביותר לתכנות במחשב הם קרוב
לודאי אלה המכילים מושגים הקרובים ביותר
לעולמו של המחשב : מספרים. אנו יכולים להשתמש
בהם בדרכים מגוונות, החל מסתם ניחוש המספר שהמחשב
ידפיס (או שחברך חושב עליו), ועד לשמוש במחשב
לשם תכנון "ריבוע קסם".
רוב המשחקים האלה כוללים, ברמה זו או אחרת, את
השמוש במספרים אקראיים ובפונקציה RND(1).
ע"י שמוש במחשב לשם ייצור מספרים אקראיים, יש
בידינו האפשרות ליצור מספר בלתי מוגבל של
משחקים המשתמשים במרכיב של סכויים. המחשב
יכול להיות הקוביה שלנו, לוח המספרים המסתובב,
השחקן היריב (או אפילו השותף במשחק), וזה שרושם
את הנקודות. יותר מכך, מאחר שאנו משתמשים
במסך הטלביזיה, יכול המחשב להיות גם לוח המשחקים.
וזה אפילו שומר על איכות הסביבה, כי אין צורך
להשתמש כל פעם בניר וחתיכות קרטון כדי ליצור משחקים
חדשים - הכל נעשה ע"י המחשב !

הבה נתחיל בכמה משחקי ניחוש פשוטים. חשוב לזכור,
שאלה הם משחקים שאפשר לתארם כ"שורש" : הם יצמחו
ויפרחו ככל שתגדל תשומת הלב שתוקדש להם. ע"י עיטורם
ועיצובם, אפשר יהיה לפתח ספרית משחקים שלמה, לשמוש
בכל פעם כדי להראות לחברים כיצד למדת להשתלט על
המחשב שלך!



נחש את המספר- ארבע גרסאות

המחשב בוחר מספר באופן אקראי ועל השחקן לנסות לנחש אותו. כדי להגדיל את מספר האפשרויות (ולכן, להיות מסוגלים "לשחק"), עלינו להגדיר למחשב תחום של מספרים. התכנית הראשונה עושה בדיוק את זה : בוחרת מספר בין 1 ו-10, ועליך לנסות לנחש מהו. זוהי אותה התכנית שהצגנו בפרק א'.

גרסה א'

```
10 REM * SIMPLE GUESS-MY-NUMBER**
20 PRINT "I HAVE PICKED A NUMBER"
25 PRINT "BETWEEN 1 AND 10."
30 PRINT "TRY TO GUESS IT."
40 LET X=INT(RND(1)*10)+1
50 PRINT "YOUR GUESS";
60 INPUT Y
70 IF Y=X THEN PRINT "YOU GOT IT!":END
80 PRINT "NO, TRY AGAIN . . .":PRINT
90 GOTO 50
```

זו הגרסה הפשוטה ביותר האפשרית. שום קישוטים- המחשב בוחר מספר בין 1 ו-10, (שורה 40) ועליך לנסות לנחש מהו. התכנית תמשיך עד שימצא המספר.

והנה שיפור פשוט של התכנית שלמעלה : היא מבקשת לדעת אם ברצונך להמשיך לשחק (כלומר, אם נחשת את המספר !). היא משתמשת בתת תכנית מסוג כן/לא שהצגנו בפרק ג'.

גרסה ב'

```
5 DIM A$(1)
10 REM ** GUESS-MY-NUMBER =2 **
15 REM ** ADD "PLAY AGAIN" FEATURE **
```

```
20 PRINT "I HAVE PICKED A NUMBER"
25 PRINT "BETWEEN 1 AND 10."
30 PRINT "TRY TO GUESS IT."
40 LET X=INT(RND(1)*10)+1
50 PRINT "YOUR GUESS";
60 INPUT Y
70 IF Y=X THEN 100
80 PRINT "NO, TRY AGAIN . . .":PRINT
90 GOTO 50
100 PRINT "YOU GOT IT!"
110 PRINT "DO YOU WANT TO PLAY AGAIN?"
120 INPUT A$
130 IF A$="N" THEN END
140 CLR
150 GOTO 20
```

גרסה ג' מוסיפה שיפור נוסף – היא מרשה לך לראות את התשובה, אם ברצונך לראותה, אחרי כל נחוש מוטעה (שורות 70–90)

גרסה ג'

```
5 DIM A$(1)
10 REM ** GUESS-MY-NUMBER =3 **
15 REM * ADD "PLAY AGAIN" FEATURE & **
17 REM * "[CLEAR} WHAT'S THE ANSWER?" **
20 PRINT "I HAVE PICKED A NUMBER"
25 PRINT "BETWEEN 1 AND 10."
30 PRINT "TRY TO GUESS IT."
40 LET X=INT(RND(1)*10)+1
50 PRINT "YOUR GUESS";
60 INPUT Y
70 IF Y=X THEN 100
80 PRINT "TYPE Y TO TRY AGAIN OR"
85 PRINT "TYPE N TO SEE THE ANSWER";
87 INPUT A$: IF A$="Y" THEN GOTO 50
90 PRINT "MY NUMBER IS";X:GOTO 110
110 PRINT "YOU GOT IT!"
110 PRINT :PRINT "DO YOU WANT TO PLAY AGAIN?"
130 IF A$="Y" THEN CLR:GOTO 20
140 PRINT "BYE FOR NOW"
150 END
```

גרסה ד' מוסיפה שיפור חשוב מאד : רמזים (שורות 140-150). ע"י כך, אפשר לדעת אם הניחוש גדול או קטן מדי, ואז ניתן להתאים את הניחוש הבא לרמז שקבלת זה עתה.

הגרסה הזו גם מאפשרת לך להרחיב את תחום המספרים שיבחר המחשב (שורות 40-50). עד עכשו, השתמשו כל הגרסאות בתחום 1-10; בגרסה זו נשתמש בתחום 1 עד MAX, כאשר MAX הוא קצה תחום המספרים על פי החלטתך. התכנית גם זוכרת כמה נחושים היו ותעצור אם לא הגעת למספר הנכון אחרי MAX ניחושים.

גרסה ד'

```
5 DIM A$(1)
10 REM ** GUESS-MY-NUMBER =4 **
20 REM ** (WITH HINTS) **
30 PRINT "I WILL PICK A NUMBER BETWEEN"
40 PRINT "1 AND (YOU FILL IN)";
50 INPUT MAX
60 PRINT "THEN, YOU TRY TO GUESS IT."
70 LET X=INT(RND(0)*MAX)+1
80 PRINT :PRINT "OK, I'VE GOT A NUMBER."
100 FOR I=1 TO MAX
110 PRINT :PRINT "YOUR GUESS";
120 INPUT Y
130 IF Y=X THEN GOTO 250
140 IF Y>X THEN PRINT "NO, THAT'S TOO HIGH. . .
    TRY AGAIN": GOTO 200
150 IF Y<X THEN PRINT "NO, THAT'S TOO LOW . . . TRY
    AGAIN"
200 NEXT I
210 PRINT :PRINT "SORRY. MY NUMBER WAS"; X
220 GOTO 290
250 PRINT :PRINT "YOU GOT IT IN"; I;" GUESSES!"
290 POP :PRINT "DO YOU WANT TO PLAY AGAIN (YES OR
    NO)";
295 INPUT A$
300 IF A$="Y" THEN CLR:GOTO 30
310 PRINT "THANKS FOR PLAYING"
399 END
```

ועכשו שכלול מעניין של משחק הניחוש הטיפוסי.
במשחק זה, המחשב מנחש את המספר שלך במקום
להפך.
מהתבוננות בדרך הפעולה של המחשב בבחירת הניחוש
שלו, אפשר ללמוד על אחת מדרכי הפעולה החשובות
של משחקי ניחוש מסוג זה !

```
10 REM << COMPUTER GUESSES YOUR # >>
20 DIM GAME(100)
30 SWITCH=53279:START=6:G=0
40 POKE SWITCH,0
50 PRINT "[CLEAR]THINK OF A NUMBER BETWE
EN"
60 PRINT "1 AND 100.  WHEN YOU'VE GOT"
70 PRINT "IT, PRESS START."
80 REM *WAIT TILL START IS PUSHED *
90 IF PEEK(SWITCH)<>START THEN 90
100 LAST=0
110 PRINT :PRINT "OK, NOW I'LL GUESS & Y
OU TYPE"
120 PRINT "  2 IF I'M TOO HIGH;"
130 PRINT "  1 IF I'M TOO LOW; &"
140 PRINT "  0 IF I'M RIGHT ON."
150 NOW=40+INT(RND(1)*20)+1
160 PRINT
170 FOR I=1 TO 10
180 PRINT "GUESS # ";I;" = ";NOW;"  (2,1
, OR 0) ";
190 TRAP 220:INPUT X
200 IF X=0 THEN 260
210 IF X=2 OR X=1 THEN GOSUB 380:GOTO 24
0
220 PRINT "PLEASE JUST TYPE 2,1, OR 0";
230 GOTO 190
240 NEXT I
250 GOSUB 480:GOTO 270
260 PRINT :PRINT "I GOT IT IN ";I;" GUES
SES."
270 PRINT :PRINT "PLAY AGAIN (1=YES, 2=N
O) ";
280 INPUT PA
290 G=G+1:GAME(G)=I
300 IF PA=1 THEN 50
310 SUM=0
```

```
320 FOR J=1 TO G
330 SUM=SUM+GAME(J)
340 NEXT J
350 PRINT :PRINT "MY AVERAGE FOR ";G;" G
AMES WAS:":PRINT :PRINT "    ";SUM/G;" GU
ESSES."
360 PRINT :PRINT "THANKS FOR PLAYING!"
370 END
380 REM * GUESS TOO HIGH OR LOW *
390 A=NOW
400 IF I=1 AND X=1 THEN LAST=100
410 DIF=ABS(NOW-LAST)
420 IF DIF<4 AND X=2 THEN NOW=NOW-1:GOTO
 460
430 IF DIF<4 AND X=1 THEN NOW=NOW+1:GOTO
 460
440 IF X=2 THEN NOW=NOW-INT(DIF/2):GOTO
460
450 IF X=1 THEN NOW=NOW+INT(DIF/2)
460 LAST=A
470 RETURN
480 REM *
490 PRINT :PRINT "ONE OF THE FOLLOWING H
AS OCCURRED:"
500 PRINT "  1. MY STRATEGY HAS FAILED"
510 PRINT "  2. YOU'RE NOT PLAYING BY TH
E RULES."
520 PRINT :RETURN
```



להגיע ל-100: שלוש גרסאות

נביא עכשו שלש גירסאות של משחק פשוט מאד : מי יהיה הראשון שיגיע ל-100 ? זה משחק לשני אנשים או יותר, וכן לשחקן ומחשב. כל שחקן בוחר מספר בתורו והמספר הזה נוסף לסכום המספרים. השחקן הראשון שמגיע למאה או עובר את המספר 100 הוא המנצח.

המשחק דומה מאד למשחקי NIM(נים) שנציג מיד. אפשר לשנות את המשחק בדרכים שונות, כמו הגדרת תחום המספרים שבתוכו מותר לבחור מספר, קביעת חוק שמוכרחים להגיע בדיוק למאה (ולא לעבור אותו), וכו'. יכולים לקשט ולעטר את המשחק עם גרפיקה עשירה (כמו לצייר בעל חיים המתקרב לקו הסיום), ברכות למנצח, ניחומים למפסיד ובדרכים שונות להעלות את המתח במשחק (פרק ב' מציע כמה דרכים כאלה).

בגרסה הראשונה של המשחק נתנו רק את המבנה הבסיסי של המשחק, בלי כל הגבלות על תחום המספרים וכמעט ללא הנחיות. הגרסה השניה מראה כיצד אפשר להרחיב את המבנה הבסיסי ולהוסיף לו הקדמה, הנחיות, והגנה כך שהשחקן יוכל לבחור רק מספר אחד בתוך תחום מספרים מוגדר (במקרה זה,מ-1 עד 10). יש גם תצוגה אור קולית למנצח. לבסוף, הגרסה השלישית מוסיפה תחרות עם המחשב, הבוחר את המספרים באופן אקראי. כל שלש הגרסאות מתירות הן להגיע בדיוק עד 100 והן לעבור את 100; אבל את זה ניתן לשנות בקלות כך שקנה המידה לנצחון יהיה המספר 100 במדויק.

להגיע ל-100: גרסה 1

```
$ REM ** GETTING TO 100:
2 REM ** THE BASIC STRUCTURE**
10 LET T=0
20 INPUT X
```

```
30 LET T=T+X
40 PRINT "TOTAL = ";T
50 IF T<100 THEN GOTO 20
60 PRINT "YOU WIN"
```

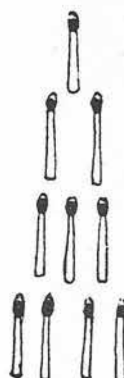
להגיע ל-100: גרסה 2

```
1 REM **GETTING TO 100:
2 REM **DRESSED UP WITH INSTRUCTIONS
3 REM ** LIMIT ON YOUR INPUT **
5 PRINT "THE OBJECT OF THIS GAME IS TO BE"
6 PRINT "THE FIRST TO GET TO 100"
7 PRINT "YOU MAY ONLY USE NUMBERS FROM 1 TO 10."
8 PRINT "EACH TIME YOU ENTER A NUMBER,"
9 PRINT "IT IS ADDED TO THE TOTAL"
10 LET T=0
12 PRINT "TYPE IN A NUMBER BETWEEN 1 AND 10";
20 INPUT X
25 IF X<1 OR X>10 THEN PRINT "NUMBER MUST BE
   BETWEEN 1 & 10";:GOTO 20
30 LET T=T+X
40 PRINT "TOTAL = ";T
50 IF T<100 THEN GOTO 12
60 PRINT "YOU WIN";
70 SETCOLOR 2, INT(RND(1)*16),4
80 SOUND 0,RND(1)*250,10,8
120 GOTO 60
```

להגיע ל-100: גרסה 3

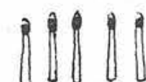
```
10 REM *       <<< GET 100 >>>
20 REM *     copyright 1980  by
30 REM *    Herb Kohl & Ted Kahn
40 DIM WIN$(7)
50 PRINT "[CLEAR]THE OBJECT OF THIS GAME
 IS TO BE"
60 PRINT "THE FIRST TO GET TO 100"
70 PRINT "YOU MAY ONLY USE NUMBERS FROM
1 TO 10."
80 PRINT "EACH TIME YOU ENTER A NUMBER,"
90 PRINT "IT IS ADDED TO THE TOTAL"
100 LET T=0
110 PRINT :TRAP 110
```

```
120 PRINT "CHOOSE YOUR NUMBER (1 THRU 10
) ";
130 INPUT GUESS
140 X=INT(GUESS)
150 IF X<1 OR X>10 THEN PRINT "NUMBER MU
ST BE BETWEEN 1 & 10 ";:GOTO 130
160 PRINT "[CLEAR]"
170 PRINT "YOU PICK ";X
180 LET T=T+X
190 PRINT "TOTAL = ";T
200 IF T>=100 THEN WIN$="YOU win":GOTO 2
70
210 C=INT(RND(1)*10)+1
220 PRINT :PRINT "I PICK ";C
230 T=T+C
240 PRINT "TOTAL = ";T:PRINT
250 IF T>=100 THEN WIN$="I win":GOTO 270
260 GOTO 120
270 GRAPHICS 2+16
280 FOR LOOP=1 TO 250
290 POSITION INT(RND(1)*14),INT(RND(1)*1
2)
300 PRINT #6;WIN$;
310 SOUND 1,RND(1)*255,6,8
320 NEXT LOOP
330 RUN
```



נים

נים הוא משחק הימורים סיני עתיק. משחקים אותו בדרך כלל בגפרורים. גירסה פשוטה של המשחק, עם שתי קבוצות של גפרורים משמשת את המהמרים לפיתוי אנשים להצטרף למשחק, הפעם בגרסאות יותר מסובכות שבהן חישוב ההימור איננו כל כך קל. לפניך הבסיס לגרסה עם שתי חבילות הגפרורים של נים. (מספר הגפרורים בכל ערמה איננו משנה).



(ערמה 1)



(ערמה 2)

על השחקנים, כל אחד בתורו, לסלק כמה גפרורים שירצו (לפחות אחד) מאחת הערמות. השחקנים יכולים להחליט בעצמם מאיזו ערמה יסלקו את הגפרורים, וכן זכותם להחליט על מטרת המשחק : זה שמוציא את הגפרור האחרון הוא המנצח, או שהמטרה היא להכריח את היריב לקחת את הגפרור האחרון.

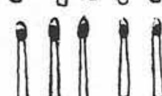
הנה דוגמה של המשחק שבו מנצח מי שלוקח את הגפרור האחרון:

הסידור הראשוני

(I)



(II)



שחקן 1 :
לוקח 3 גפרורים
מערמה II

(I)



(II)



שחקן 2 :
לוקח 3 גפרורים
מערמה I

(I)



(II)



שחקן 1 :
לוקח גפרור אחד
מערמה I

(I)



(II)



שחקן 2 :
לוקח גפרור אחד
מערמה II , ומנצח
כי שחקן 1 יכול לקחת
גפרור רק ממקום אחד.

(I)



(II)



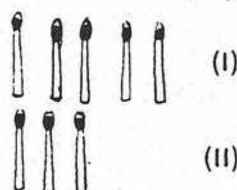
(I)

שחקן 1 :

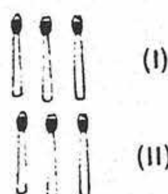
(II) לוקח גפרור מערמה I

ושחקן 2 לוקח את הגפרור האחרון.

בגרסה הבאה של משחק בשתי קבוצות גפרורים, השיטה היא לאלץ את היריב לקחת גפרורים משתי ערימות שוות.
השחקן יכול לשמור על מספר שווה של גפרורים בשתי הקבוצות עד שהוא נשאר האחרון לקחת את הגפרור האחרון :

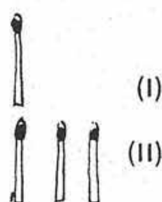


אם השחקן הראשון לוקח שני גפרורים מ I, ומשווה את שתי הקבוצות, הוא יכול לכפות הפסד על היריב.

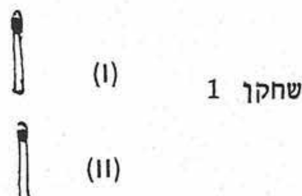


אחרי מהלך של שחקן 1

תורו של שחקן 2 - הוא לוקח שנים מקבוצה I



ושחקן 1 מגיב ע"י השואת הקבוצות וכופה נצחון (ע"י כך שהוא מבטיח לעצמו את הגפרור האחרון).



שחקן 1

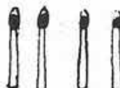
שחקן 2 (I)

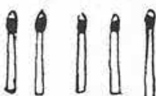
 (II)

שחקן 2 לוקח את הגפרור האחרון ומנצח.

משחק נים נעשה הרבה יותר מסובך כאשר משחקים בשלש או יותר ערמות גפרורים. הנה משחק שבו יש שלש ערמות. החוקים זהים. כדאי לנסות כמה פעמים, לפני שנכנסים לעולם היותר מורכב של נים, באמצעות המחשב.

(I) 

(II) 

(III) 

השחקן שיקח את הגפרור האחרון הוא המנצח.

נים הוא משחק חיסור וגירסה פשוטה ביותר עבור המחשב יכולה להכתב בחמש שורות בלבד.

```
10 INPUT X
20 INPUT Y
30 X=X-Y
40 IF X>0 THEN GOTO 20
50 PRINT "YOU WON"
```

בתכנית הזו X הוא המספר ההתחלתי של המשחק ו-Y הוא המספר שאותו צריכים להפחית במירוץ לכוון 0.
מטרת המשחק (שאותו יכולים לשחק בכל מספר שחקנים, אם כי שנים או יותר הוא הטוב ביותר כנראה), הוא להיות השחקן שמגיע ל-0. השחקן יכול להיות 1 או 2.



שורה 30 מורידה את ערכו של X כל פעם ששחקן מכניס ערך עבור Y, ושורה 40 ממשיכה את המשחק עד שמגיעים ל-0. כשמגיעים ל-0, מורה שורה 50 למחשב להדפיס "YOU WON" (נצחת). השחקנים מתחלפים בתורם, ומכניסים 1 או 2.

הנה גירסה "מעוטרת" מעט של המשחק, אשר מוסיפה הנחיות ומאפשרת לך לדעת מה גדלו של X אחרי מהלך של כל שחקן :

```
10 REM SIMPLE NIM WITH NUMBERS
20 PRINT "YOU START WITH X"
30 INPUT X
40 PRINT "TAKE AWAY 1 OR 2"
50 PRINT "PLAYER WHO TAKES LAST NUMBER WINS"
60 PRINT "WHICH WILL YOU TAKE AWAY,1 OR 2"
70 INPUT Y
80 X=X-Y
85 PRINT "NUMBER LEFT",X
90 IF X>0 THEN GOTO 60
100 IF X=0 THEN PRINT "YOU WON"
```

אפשר גם, תוך שימוש במושגים שהצגנו כבר בחלק הראשון לתכנת גרסה חזותית של המשחק הפשוט הזה. בגרסה כאן 10 הוא המספר ההתחלתי, אולם ע"י שנוי המשתנים אפשר לעלות עד 75 בלי לגלוש מהמסך במצב גרפי 7.

```
10 REM THIS GAME INVOLVES TAKING ONE
11 REM OR TWO AWAY AND SEEING THE GAME
12 REM ON THE SCREEN
13 REM ANY NUMBER OF DOTS CAN BE USED
14 REM AS LONG AS THEY CONFORM TO GRAPHICS
20 GRAPHICS 7:COLOR 1
30 LET A=15
40 LET W=10
50 FOR X=1 TO 10
60 PLOT 10*X,A
70 NEXT X
80 PRINT "Y=?":INPUT Y
91 LET A=A+5
100 LET W=W-Y
110 FOR Z=1 TO W
120 PLOT 10*Z,A
130 NEXT Z
140 IF W=0 THEN GOTO 200
150 GOTO 80
200 PRINT "NIM":END
```

באירופה משחקים נים בגפרורי עץ, בדרך כלל, ע"י GRAPHICS 3, אפשר לעטר את הגרסה האחרונה שלנו של

נים בצבעים. הנה תת תכנית פשוטה שמצירת עד 15
גפרורים על המסך.כדאי לנסות אותה כמה פעמים.

```
1000 REM ROUTINE TO DRAW MATCHES
1010 GRAPHICS 3
1012 PRINT "HOW MANY MATCHES (1-15)";
1014 INPUT N
1020 FORK=1 TO N*2 STEP 2
1030 COLOR 1
1035 PLOT K5
1040 COLOR 2
1050 DRAWTO K,10
1060 NEXT K
```

והנה גרסה מלאה של נים עבור שני שחקנים, תוך
שמוש בהצגה גרפית מענינת. התכנית הזו נכתבה ע"י
דייל דישרון.



```
0 REM *    <<< NIM >>>
10 REM * COPYRIGHT 1981
20 REM * by Dale Disharoon
30 GRAPHICS 0:SETCOLOR 2,0,0
40 POSITION 14,4:PRINT " << NIM >> "
50 POSITION 2,8:PRINT " How high will yo
ur ":PRINT " pyramid be? (3-10) "
60 TRAP 60:POSITION 25,9:INPUT HIGH
70 IF HIGH<3 OR HIGH>10 THEN 60
80 POSITION 2,15:PRINT " 1 - Player taki
ng last block wins  "
90 PRINT " 2 - Player taking last block
loses "
100 TRAP 100:POSITION 11,18:PRINT " Ente
r 1 or 2 ";:INPUT LAST
110 IF LAST<1 OR LAST>2 THEN 100
120 PRINT "[CLEAR]":POKE 752,1
130 POSITION 14,1:PRINT " << NIM >> "
140 DIM X(HIGH,HIGH),Y(HIGH,HIGH),B(10)
150 FOR H=1 TO HIGH
160 B(H)=H
170 FOR W=1 TO H
180 READ X:X(H,W)=X:Y(H,W)=H+4
190 POSITION X(H,W),Y(H,W):PRINT CHR$(16
0);CHR$(130);
```

```
200 NEXT W:PRINT " ";H:NEXT H
210 DATA 11,10,12,9,11,13,8,10,12,14,7,9
,11,13,15,6,8,10,12,14,16,5,7,9,11,13,15
,17,4,6,8,10,12,14,16,18
220 DATA 3,5,7,9,11,13,15,17,19,2,4,6,8,
10,12,14,16,18,20
230 POSITION 23,6:PRINT "* Last block *"
:POSITION 28,7
240 IF LAST=1 THEN PRINT "Wins"
250 IF LAST=2 THEN PRINT "Loses"
260 P=P+1:IF P>2 THEN P=1
270 TRAP 270:POSITION 2,17:PRINT "[DEL L
INE] Player #";P;" takes from level-";:I
NPUT H
280 IF H<1 OR H>HIGH THEN PRINT "[UP][DE
L LINE]Sorry, there is no level ";H:FOR
T=1 TO 500:NEXT T:GOTO 270
290 IF B(H)=0 THEN PRINT "[UP][DEL LINE]
Level ";H;" is gone":FOR T=1 TO 500:NEXT
 T:GOTO 270
300 TRAP 300:POSITION 2,19:PRINT "[DEL L
INE] Player #";P;" takes how many-";:INP
UT N
310 IF N<1 THEN PRINT "[UP][DEL LINE]I c
an't let you do that":FOR T=1 TO 500:NEX
T T:GOTO 300
320 IF N>B(H) THEN PRINT "[UP][DEL LINE]
There aren't that many in level ";H:FOR
T=1 TO 500:NEXT T:GOTO 300
330 B(H)=B(H)-N
340 FOR T=B(H)+1 TO B(H)+N:POSITION X(H,
T),Y(H,T):PRINT "  ":GOSUB 380:NEXT T
350 POSITION 2,17:PRINT "[DEL LINE][DOWN
][DEL LINE]"
360 GOSUB 400
370 GOTO 260
380 FOR X=21 TO 0 STEP -3:SOUND 0,X,10,8
:NEXT X:SOUND 0,0,0,0:RETURN
390 FOR X=50 TO 0 STEP -2:SOUND 0,X,10,8
:SOUND 1,X+1,10,8:NEXT X:SOUND 0,0,0,0:S
OUND 1,0,0,0:RETURN
400 FOR T=1 TO HIGH:IF B(T) THEN POP :RE
TURN
410 NEXT T:POSITION 5,17
```

```
420 IF LAST=1 THEN FOR T=1 TO 5:GOSUB 39
0:NEXT T:PRINT "____xxx__ Player #";P;" wi
ns! _xxx____":GOTO 450
430 SOUND 0,5,4,8:FOR T=1 TO 100:NEXT T:
SOUND 0,0,0,0
440 PRINT "____xxx__ Player #";P;" loses _
xxx____"
450 OPEN #1,4,0,"K:"
460 POSITION 14,22:PRINT "Press RETURN"
470 GET #1,T:RUN
```

קוביות

קל מאד לחקות הטלת קוביות באמצעות פונקצית RND(1) של האטארי. (RND(1)*6)+1 תייצר מספר אקראי בין 1 ו-6, בדיוק כמו בהטלת קוביה. (RND(1)*12)+1 תתן אותה תוצאה כמו הטלת שתי קוביות. איזו פונקצית RND(1) תתן אותה התוצאה כמו הטלת 4 קוביות ?

הנה תכנית פשוטה שמדפיסה מספרים הנובעים מבחירה אקראית של מספרים מ-1 עד 6, ואשר זהים להטלת קוביה אחת :

```
10 - R=INT(RND(1)*6)+1
20 - PRINT R
30 - GOTO 10
```

שנוי קל יביא לכך שהמחשב ידפיס נקודות (כפי שאפשר לראות על הקוביה), במקום מספרים :

```
5 REM DICE1 (FIRST REPRESENTATION): THROW ONE DIE
  & SHOW THE RESULT
6 REM (C) BY TED M. KAHN, 1980
10 R=INT(RND(1)*6)+1
20 FOR I=1 TO R
30 PRINT "[T]";
40 NEXT I
45 PRINT
50 GOTO 10
```

צריך לזכור : (T) פרושו ללחוץ ולהחזיק את מקש (CTRL) ובאותו זמן לתקתק את האות " T ". דבר זה יתן את הסימן הגרפי (●)

והנה שכלול נוסף שבו מטילים שתי קוביות פעם אחת והשחקן יכול לבחור אם להטיל אותן שנית, או לא.

```
5 REM DICE2 (SECOND REPRESENTATION): THROW TWO
  DICE & SHOW THE RESULT
6 REM (C) BY TED M. KAHN, 1980
8 DIM Y$(1)
10 R1=INT(RND(1)*6)+1
15 R2-INT(RND(1)*6)+1
20 FOR I=1 TO R1
30 PRINT "[T]";
40 NEXT I
45 PRINT ,
50 FOR J=1 TO R2
60 PRINT "[T]";
70 NEXT J
80 PRINT :PRINT
90 PRINT "THROW AGAIN";
95 INPUT Y$:IF Y$="Y" THEN PRINT :GOTO 10
100 END
```

בדפים הבאים אנו מביאים כמה משחקים מורכבים עם גורם של אקראיות והימור, ואשר בהם בוחר המחשב באופן אקראי ומחקה את הטלת הקוביה.



שלוש קוביות

משחקי מזל והימורים הם מן המשחקים העתיקים ביותר הידועים לנו, והם עדיין מקובלים ברוב התרבויות בעולם כיום.

תוך שמוש במושגים שהצגנו לגבי מספרים אקראיים, (INT(RND(1)*6)+1), משחק זה שלפניך מחקה הטלת שלש קוביות ומראה את התוצאות דרך המצב הגרפי 3 של אטארי. למעשה, תת התכנית בשורות 870-1080 (יחד עם הכרזת DIM בשורה 20), נתנת לשינוי לשימוש במשחקים אחרים שבהם יש צורך בתת תכנית שתטיל קוביות.
המשחק גם משתמש בעטורים משוכללים שתוכננו ע"י לן לינדסי, כדי לתת להקדמה מראה של שלט ניאון מהבהב או שלט פרסומת צבעוני. האפקטים הללו נוצרים בשורות 130-220 ומשתמשים באותיות הגדולות שבאפשרותנו לכתוב במצבים GRAPHICS 2+16 (מסך מלא).

למעשה, זהו משחק הימורים. מתחילים עם מספר מסוים של נקודות, נניח 100 והשחקן יכול להמר ככל שהוא רוצה על אחד המספרים (1-6). אם המספר מופיע על <u>לפחות</u> קוביה אחת, השחקן מכפיל את ההימור. אם המספר מופיע על שתי קוביות, השחקן משליש את ההימור ; ואם המספר מופיע על כל שלש הקוביות (הסכויים הם 1 מתוך 216), השחקן מנצח.

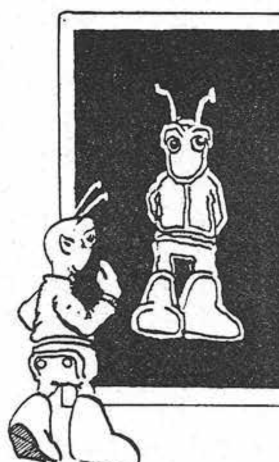
```
10 REM *     <<< THRICE DICE >>>
20 REM *   (C) 1980 by Len Lindsay
30 DIM DX(2),DY(2),N(2),A$(6)
40 CS=100
50 MAXCS=900000
60 COL=1
70 GRAPHICS 2+16:PRINT #6;"THRICE DICE"
80 POSITION 0,2:PRINT #6;"TO BEGIN"
90 PRINT #6;"HIT start"
100 POSITION 0,7:PRINT #6;"FOR INSTRUCTI
ONS"
110 PRINT #6;"HIT option"
120 POSITION 2,11:PRINT #6;"(C) 1980"
130 KEY=PEEK(53279)
140 IF KEY=3 THEN GOSUB 630:GOTO 70
150 IF KEY=5 THEN CS=CS+100:IF CS>1000 T
HEN CS=CS+900:IF CS>10000 THEN CS=CS+900
0:IF CS>100000 THEN CS=CS+90000
160 IF CS>MAXCS THEN CS=100
170 IF KEY=6 THEN 230
180 POSITION 0,5:PRINT #6;"CASH=$";CS;"
select   "
190 SETCOLOR 1,0,RND(1)*256
200 FOR DELAY=1 TO 99:NEXT DELAY
210 PN=12-PN:SETCOLOR 2,0,PN*RND(1)*256
220 GOTO 130
230 GRAPHICS 3
240 SETCOLOR 2,0,0
250 PRINT "[CLEAR]";
260 PRINT "YOU HAVE $";CS;" CASH."
270 IF CS>999999 THEN 1140
280 PRINT "HOW MUCH WILL YOU BET";
290 TRAP 250
300 POKE 764,255
310 INPUT A$
```

```
320 IF A$="H" OR A$="HELP" THEN GOSUB 63
0:GOTO 230
330 BET=INT(VAL(A$))
340 IF BET<1 THEN 250
350 IF BET>CS THEN 250
360 PRINT "WHAT NUMBER WILL YOU BET ON";
370 TRAP 360
380 POKE 764,255
390 INPUT A$
400 IF A$="H" OR A$="HELP" THEN GOSUB 63
0:GOTO 230
410 CS=CS-BET
420 PRINT "[CLEAR]"
430 NUM=INT(VAL(A$))
440 IF NUM=0 OR NUM>6 THEN 370
450 TRAP 34567
460 FOR A=0 TO RND(1)*10
470 GOSUB 870
480 PRINT "[CLEAR]YOU BET $";BET;" ON ";
NUM;".    ";
490 NEXT A
500 MATCH=0
510 FOR A=0 TO 2
520 IF NUM=N(A) THEN MATCH=MATCH+1
530 NEXT A
540 WIN=0
550 IF MATCH>0 THEN WIN=BET*(MATCH+1)
560 CS=CS+WIN
570 PRINT MATCH;" MATCH."
580 PRINT "YOU GET $";WIN;" CASH BACK!"
590 IF CS<1 THEN 1130
600 GOTO 260
610 GOTO 610
620 END
630 GRAPHICS 1+16
640 PRINT #6;"THIS IS A GAMBLING"
650 PRINT #6;"DICE GAME. YOU BET "
660 PRINT #6;"MONEY ON ONE OF THE"
670 PRINT #6;"6 POSSIBLE NUMBERS."
680 PRINT #6
690 PRINT #6;"if none of the dice"
700 PRINT #6;"show that number "
710 PRINT #6;"you lose the bet."
720 PRINT #6
```

```
730 PRINT #6;"IF 1 MATCHES YOU"
740 PRINT #6;"GET DOUBLE YOUR BET"
750 PRINT #6;"BACK (5 BET WINS 10)"
760 PRINT #6;"if two dice match"
770 PRINT #6;"you get triple back"
780 PRINT #6
790 PRINT #6;"-IF ALL DICE MATCH"
800 PRINT #6;"YOU GET 4 TIMES BACK"
810 PRINT #6;"SOUNDS GOOD - HUH!!"
820 PRINT #6
830 PRINT #6
840 PRINT #6;"hit start to return";
850 GOSUB 1090:IF KEY<>6 THEN 850
860 RETURN
870 FOR D=0 TO 2
880 N(D)=INT(RND(1)*6)+1
890 NEXT D
900 FOR D=0 TO 2
910 DX(D)=INT(RND(1)*6)+13*D
920 DY(D)=INT(RND(1)*13)
930 NEXT D
940 PRINT #6;"[CLEAR]"
950 FOR D=0 TO 2
960 COL=3-COL:COLOR COL
970 FOR R=DY(D) TO DY(D)+6
980 PLOT DX(D),R:DRAWTO DX(D)+6,R
990 NEXT R
1010 COLOR 0
1020 FOR D=0 TO 2
1030 IF N(D)>1 THEN PLOT DX(D)+1,DY(D)+1
:PLOT DX(D)+5,DY(D)+5
1040 IF N(D)>3 THEN PLOT DX(D)+1,DY(D)+5
:PLOT DX(D)+5,DY(D)+1
1050 IF N(D)>5 THEN PLOT DX(D)+1,DY(D)+3
:PLOT DX(D)+5,DY(D)+3
1060 IF N(D)/2>INT(N(D)/2) THEN PLOT DX(
D)+3,DY(D)+3
1070 NEXT D
1080 RETURN
1090 IF PEEK(53279)<>7 THEN 1090
1100 KEY=PEEK(53279):IF KEY=7 THEN 1100
1110 IF PEEK(53279)<>7 THEN 1110
1120 RETURN
1130 PRINT "YOU HAVE NO MONEY LEFT":GOTO 1150
```

```
1140 PRINT "YOU BROKE THE BANK!!!"
1150 PRINT "HIT START TO PLAY AGAIN.";
1160 GOSUB 1090
1170 IF KEY<>6 THEN 1160
1180 RUN
```

זוהי תכנית מסובכת, ולא קל להעתיק אותה. ברגע שתעבוד, רצוי להעתיק אותה על סרט או דיסקט, אלא אם כן עומד לרשותך זמן רב להעתיק אותה כל פעם מחדש.



רברס

רברס הומצא לפני כמה שנים ע"י פיטר ל. סשונז מחברת People's Computer. הוא פורסם לראשונה בעתון PCC (שהפך אחר כך ל People's Computers, Recreational Computing.)
וכיום מהווה חלק מהעתון Compute.
המשחק נעשה פופולארי במיני מחשבים ובמחשבים זעירים. הגרסה שנביא כאן תוקנה ע"י אחד המחברים כדי להראות כיצד מחשב ביתי כמו אטארי 400 או אטארי 800 יכול, עם צבע, קול ותנועה, לשפר גם משחק שהיה טוב מלכתחילה.

במשחק **רברס** מתמרנים מחרוזת של ספרות כך שהמחרוזת הסופית תהיה מסודרת בסדר עולה. המשחק מתחיל כשהמחשב מציג סדרה של מספרים חד ספרתיים, ערוכים שלא לפי

סדר ; כל ספרה מופיעה רק פעם אחת, ואורך המחרוזת יכול להשתנות בגרסאות שונות של המשחק (ככל שהמחרוזת ארוכה יותר, כך המשחק יותר קשה). כך, למשל, נניח שהיתה לנו סדרה של 5 ספרות, כזו :

34215

המטרה שלנו היא שלבסוף יהיה 12345. כדי להגיע לכך, הצעד החוקי היחיד שמותר לנו לבצע הוא להפוך בין הספרות במספר מסוים מבין הספרות הללו, כשתמיד מתחילים מהקצה השמאלי של המחרוזת. כל היפוך כזה משנה את המחרוזת, ואנו מקוים, שגם מקרב אותנו למטרה. עם הדוגמה שלפנינו, נוכל להמשיך את המשחק כך :

34215	:	המספר המקורי.
34215	:	רברס 2 (כלומר, צריך לקחת את שתי הספרות הראשונות, ולהפוך את הסדר ומקבלים 43215).
43215	:	רברס 4 (להפוך את הסדר של 4 הספרות הראשונות).
12345	:	ונצחנו בשני תורות בלבד !

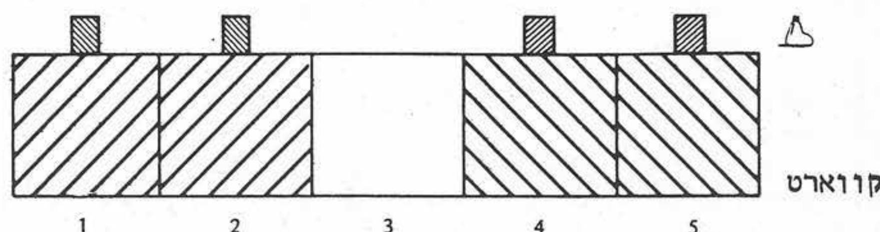
כן, זה היה די קל, אבל העניין נעשה יותר קשה עם מחרוזות יותר ארוכות. המשחק הקלסי הוא עם תשע ספרות (1-9). אפשר להוכיח באופן מתמטי שאפשר לסדר בסדר עולה מחרוזת בת 9 ספרות ב-15 צעדים (2n-3), כאשר n הוא מספר הספרות במחרוזת; אבל, לפי מיטב ידיעתנו לא הצליח אף אחד עד היום לפתח דרך שתמצא את המספר הקטן ביותר של צעדים עבור כל מחרוזת בת 9 ספרות שהיא. אולי תהיו את, או אתה, הראשונים ! והנה התכנית למשחק :

```
10 REM *        <<< REVERSE >>>
20 REM *     Adapted by Ted Kahn
30 REM *  from P.L. Sessions P.C.C.
40 DIM NAME$(20),A$(10)
50 DIM XX$(1)
60 DIM ELIST$(9),MYLIST$(9),RLIST$(9)
70 NAME$="REVERSE"
80 GOSUB 290
90 OPEN #1,4,0,"K:"
100 PRINT "WELCOME TO THE GAME OF REVERS
E"
110 PRINT "DO YOU NEED INSTRUCTIONS?";:G
```

```
OSUB 870:IF XX=41 THEN GOSUB 1000
120 PRINT "[CLEAR]HOW MANY DIGITS WOULD
YOU":PRINT "LIKE (3-9)";
130 GOSUB 870:MAX=XX:REM INPUT
140 PRINT XX;
150 IF MAX<3 OR MAX>9 THEN PRINT "[CLEAR
]SORRY...ONLY NUMBERS BETWEEN 3-9":GOTO
120
160 GOSUB 420:REM INITIAL SEQUENCE
170 TURN=TURN+1
180 PRINT "[CLEAR]"
190 PRINT "TURN # ";TURN
200 PRINT :PRINT "REVERSE HOW MANY? ";
210 GOSUB 870:REM INPUT
220 R=XX:PRINT R;
230 FOR WW=1 TO 200:NEXT WW
240 IF R<2 OR R>MAX THEN 270
250 GOSUB 620
260 GOTO 170
270 PRINT "CAN ONLY REVERSE FROM 2 - ";M
AX;"  DIGITS":FOR WW=1 TO 1200:NEXT WW
280 GOTO 180
290 X=12
300 GRAPHICS 2
310 FOR I=1 TO LEN(NAME$)
320 POSITION X+1,3
330 PRINT #6;NAME$(LEN(NAME$)-I+1,LEN(NA
ME$))
340 SOUND 0,INT(RND(0)*255)+1,10,12
350 FOR WW=1 TO 80:NEXT WW
360 X=X-1
370 NEXT I
380 SOUND 0,0,0,0
390 POSITION 9,5:PRINT #6;"by"
400 POSITION 6,7:? #6;"ted kahn"
410 RETURN
420 FOR K=1 TO MAX
430 MYLIST$(K,K)="0":ELIST$(K,K)="0"
440 NEXT K
450 FOR I=1 TO MAX
460 RN=INT(RND(0)*MAX)+1
470 IF ELIST$(RN,RN)<>"0" THEN 460
480 MYLIST$(I,I)=STR$(RN):ELIST$(RN,RN)=
STR$(RN)
```

```
490 NEXT I
500 GRAPHICS 2
510 X=9+INT(MAX/2)
520 FOR I=1 TO MAX
530 GRAPHICS 2
540 POSITION X,4
550 PRINT #6;MYLIST$(1,I)
560 SOUND 1,INT(RND(0)*100)+(2*I),10,10:
WW=5:GOSUB 690
570 SOUND 1,0,10,10:WW=5:GOSUB 690
580 X=X-1
590 NEXT I
600 SOUND 1,0,0,0
610 RETURN
620 FOR S=1 TO R
630 RLIST$(R-S+1,R-S+1)=MYLIST$(S,S)
640 NEXT S
650 MYLIST$(1,R)=RLIST$(1,R)
660 IF MYLIST$=ELIST$ THEN 720
670 GOSUB 500
680 RETURN
690 REM WAIT LOOP
700 FOR WAIT=1 TO WW:NEXT WAIT
710 RETURN
720 FOR I=1 TO 9
730 GRAPHICS 2:POSITION 5,4:PRINT #6;MYL
IST$
740 SETCOLOR 4,INT(RND(0)*16),8
750 FOR Q=15 TO 30 STEP 0.5
760 SOUND 0,Q,10,10
770 NEXT Q:SOUND 0,0,0,0
780 NEXT I
790 PRINT "[CLEAR]YOU WON IN ";TURN;" TU
RN";:IF TURN<>1 THEN PRINT "S";
800 PRINT "!":SETCOLOR 4,0,0
810 TRAP 840:PRINT "WOULD YOU LIKE TO PL
AY AGAIN (Y,N)";
820 INPUT A$:IF A$(1,1)="N" THEN 840
830 TURN=0:POP :? "[CLEAR]":GOTO 120
840 PRINT "THANKS FOR PLAYING!":END
870 REM  INPUT SUBROUTINE
880 SOUND 0,200,10,8:FOR WS=1 TO 100
890 NEXT WS:SOUND 0,0,0,0:POKE 764,255
900 GET #1,XX:XX=XX-48
```

```
910 TRAP 920
920 TRAP 34567
930 RETURN
1000 POKE 752,1:PRINT "[CLEAR]YOU WILL B
E GIVEN SOME SCRAMBLED":PRINT "NUMBERS F
ROM 1 TO THE NUMBER YOU"
1010 PRINT "CHOOSE. ( 3 - 9 )  PRESS RET
URN":GOSUB 870
1020 PRINT "[CLEAR]BY REVERSING BLOCKS O
F NUMBERS TRY":PRINT "TO UNSCRAMBLE THEM
 SO THEY ARE"
1030 PRINT "IN THE USUAL ORDER OF 1,2,3,
4,5 ETC.":PRINT "PRESS RETURN";:GOSUB 87
0
1040 PRINT "[CLEAR]WHEN YOU REVERSE A BL
OCK OF NUMBERS":PRINT "THE REVERSAL ALWA
YS STARTS FROM THE"
1050 PRINT "LEFT.  FOR EXAMPLE...  PRESS
 RETURN":GOSUB 870
1060 PRINT "[CLEAR]IF YOU REVERSE 3 NUMB
ERS OF THIS":PRINT "COMBINATION: 6421735
   YOU WOULD GET"
1070 PRINT "2461735.  ONLY THE 6,4, AND
2 ARE":PRINT "REVERSED.  READY?  PRESS
 RETURN";:GOSUB 870:RETURN
```



מטרת המשחק קוארט היא להחליף את צבע הקופסאות שברבועים 1 ו-2 באלה של הרבועים 4 ו-5.
יש שני צעדים שמותר לבצע :

1. להזיז אחת מן הקופסאות לריבוע <u>הריק</u> ע"י תקתוק מספר הקופסה שאותה רוצים להזיז ;
2. לקפוץ <u>מעל</u> קופסה <u>אל תוך</u> ריבוע ריק, גם כן ע"י תקתוק מס' הקופסה שאותה רוצים להזיז.

△ הריבועים הקטנים בראש הציור מזהים את המטרה של החלפת הצבעים מתחת לרבועים.

עם מעט תכנון, אפשר לעשות זאת ב-8 צעדים. קוראים המתעניינים במתמטיקה ירצו אולי להוכיח זאת לעצמם, ועליהם להשוות את המשחק הזה עם החידות המסורתיות "מגדלי האנוי"או " מיסיונרים וקניבלים".

```
10 REM *        <<< QWERT >>>
20 REM *   developed by Len Lindsay
30 REM *   and modified by Ted Kahn
40 REM * Copyright 1980 by Len Lindsay
50 GOSUB 930
60 DIM A$(10),PA$(1),P(7)
70 DIM XX$(1)
80 GRAPHICS 3
90 SETCOLOR 1,5,8
100 P(0)=3
110 P(1)=1
120 P(2)=1
130 P(3)=0
140 P(4)=2
150 P(5)=2
160 P(6)=3
170 P(7)=3
180 GOSUB 490:REM DRAW BOARD
190 MV=0:REM MOVES
200 GOSUB 590:REM DRAW PIECES
210 IF I>0 THEN 250
220 ? "[CLEAR]":? "THIS IS THE GAME OF Q
WERT"
230 ? :? "WOULD YOU LIKE INSTRUCTIONS (Y
,N)";:GOSUB 970:REM GET
240 IF XX$="Y" THEN GOSUB 760
250 PRINT "[CLEAR]";
260 PRINT "   1      2      3      4      5"
:PRINT "MOVE #";MV+1
270 TRAP 250
280 GOSUB 970:REM GET
290 M=XX
300 IF M<1 OR M>5 THEN ? "ILLEGAL MOVE--
TRY AGAIN":GOSUB 920:GOTO 260
310 REM
320 P=P(M):REM PIECE TO MOVE?
330 IF P=0 THEN ? "CAN'T DO THAT--TRY AG
AIN":GOSUB 920:GOTO 260
340 GOSUB 370:REM CHECK MOVE & MOVE IT
```

```
350 GOTO 250:REM NEXT MOVE
360 END
370 IF P(M+1)=0 THEN P(M+1)=P:P(M)=0:GOT
0 440
380 IF P(M+2)=0 THEN P(M+2)=P:P(M)=0:GOT
0 440
390 IF P(M-1)=0 THEN P(M-1)=P:P(M)=0:GOT
0 440
400 IF M>1 THEN IF P(M-2)=0 THEN P(M-2)=
P:P(M)=0:GOTO 440
410 PRINT "YOU CAN'T MOVE THAT PIECE"
420 FOR WW=1 TO 300:NEXT WW
430 RETURN
440 MV=MV+1
450 GOSUB 590:REM DRAW PIECES
460 IF P(1)=2 AND P(2)=2 AND P(4)=1 AND
P(5)=1 THEN 680:REM WINNER
470 IF MV>22 THEN SETCOLOR 4,12,4:PRINT
"[CLEAR]YOU SEEM LOST":GOTO 720
480 RETURN
490 REM DRAW BOARD
500 COLOR 3
510 PLOT 2,12:DRAWTO 32,12
520 FOR X=2 TO 32 STEP 6
530 FOR Y=13 TO 17
540 PLOT X,Y
550 NEXT Y
560 NEXT X
570 PLOT 2,18:DRAWTO 32,18
580 RETURN
590 FOR PP=3 TO 27 STEP 6
600 COLOR P((PP+3)/6)
610 FOR Q=13 TO 17
620 PLOT PP,Q:DRAWTO PP+4,Q
630 NEXT Q
640 NEXT PP
650 COLOR 1:PLOT 23,11:PLOT 29,11
660 COLOR 2:PLOT 5,11:PLOT 11,11
670 RETURN
680 PRINT "[CLEAR]"
690 PRINT "YOU WON !!!";
700 SETCOLOR 4,4,4
710 IF MV=8 THEN PRINT "-PERFECT PLAYING
!"
```

```
720 PRINT "- TRY AGAIN?"
730 GOSUB 970:REM GET
740 IF XX$="Y" THEN 80
750 GRAPHICS 0:END
760 ? "[CLEAR]The object of this game is
 to switch"
770 ? "the boxes on the left with the bo
xes"
780 ? "on the right in as few moves as"
790 ? "possible ( press RETURN to contin
ue)";:GOSUB 970
800 ? "[CLEAR] 1      2      3      4
   5":? "The boxes are numbered as shown
 here"
810 PRINT "HIT RETURN TO CONTINUE";:GOSU
B 970
820 ? "[CLEAR]A box may be moved to an e
mpty space"
830 ? "by typing its current number."
840 ? "(RETURN to continue)";:GOSUB 970
850 ? "[CLEAR]You may move either"
860 ? "(1) into an empty neighboring spa
ce"
870 ? "or (2) jump over 1 box into an "
880 ? "empty space ( RETURN to continue)
";:GOSUB 970
890 ? "[CLEAR]Try to do it in as few as
8 moves!"
900 ? "GOOD LUCK! (Press RETURN to start
)";:GOSUB 970
910 RETURN
920 FOR WW=1 TO 100:NEXT WW:RETURN
930 GRAPHICS 1:POSITION 8,4:PRINT #6;"QW
ERT"
940 POSITION 4,6:PRINT #6;" (C) 1980 BY"
:? #6;"     LEN LINDSAY   ":? #6;"
   &":? #6;"      TED KAHN"
950 FOR WW=1 TO 99:NEXT WW
960 RETURN
970 REM  'GET' SUBROUTINE
980 SOUND 0,180,10,8
990 FOR S=1 TO 100:NEXT S
1000 SOUND 0,0,0,0
1010 POKE 764,255
```

```
1020 TRAP 1020
1030 CLOSE #4
1040 OPEN #4,4,0,"K:"
1050 GET #4,XX
1060 CLOSE #4
1070 XX$=CHR$(XX)
1080 TRAP 1110
1090 XX=0
1100 XX=VAL(XX$)
1110 TRAP 34567
1120 RETURN
```

והנה עוד שלש תכניות שאפשר להעתיק ולשחק. המשחק הראשון הוא גרסה של המשחק המפורסם בינגו. המשחק השני, פאנטום, הוא משחק מסתורין שמערבב אותיות ומספרים. המשחק השלישי, "בריין באסטר" (BRAIN BUSTER) הוא גרסה יותר מורכבת של קוארט שראינו קודם.
ההנחיות לכל שלשת המשחקים כלולות בתכניות עצמן, (כדאי לשים לב שהמשחק השלישי דורש שמוש בידית בקרה,"ג'ויסטיק")

```
10 REM *      <<< BINGO >>>
20 REM *    Copyright 1980 by
30 REM *        Len Lindsay
40 GRAPHICS 2
50 DIM LINE$(40),SPACE$(40),PART$(5),B$(
2),I$(2),N$(2),G$(2),O$(2):REM ONLY NEED
ED FOR CARD MAKER
60 LMARG=2:POKE 82,LMARG
70 DIM N(75)
80 DIM BINGO$(5)
90 DIM XX$(3)
100 DIM PICK(5)
110 BINGO$="BinGo"
120 SETCOLOR 2,14,4
130 PRINT #6;"        HIT start"
140 PRINT #6;"        when ready"
150 PRINT #6;"         TO PLAY"
160 PRINT #6;"          bingo?"
170 IF PEEK(53279)<>6 THEN SOUND 0,RND(1
)*255,10,15:GOTO 170
180 SOUND 0,0,0,0
190 GRAPHICS 2
200 PRINT #6;"wait please"
210 GOSUB 7000:REM INIT ARRAY
```

```
220 PRINT "HOW MANY BINGO CARDS DO YOU W
ANT"
230 PRINT "PRINTED (HIT RETURN FOR NONE)
?"
240 GOSUB 15000
250 IF XX=0 THEN 270:REM NO CARDS PRINTE
D
260 FOR CARDPRINT=1 TO XX:GOSUB 10000:NE
XT CARDPRINT
270 NP=0
280 PRINT #6,"shuffling"
290 GOSUB 7100:REM RANDOMIZE
300 NP=NP+1
310 IF NP=76 THEN PRINT "COME NOW! EVERY
 NUMBER IS CALLED":GOTO 3010
320 NM=N(NP):REM PICK NUMBER
330 GRAPHICS 2:PRINT #6;" THE number is:
"
340 PRINT #6
350 PRINT #6," ";
360 GOSUB 4000:REM PRINT THE NUMBER
370 SETCOLOR 2,5,4
380 PRINT "[CLEAR]HIT 'B' IF SOMEONE HAS
 BINGO"
390 PRINT "HIT RETURN FOR THE NEXT NUMBE
R"
400 GOSUB 15100:REM GET
410 IF XX$="B" THEN 1000
420 GOTO 300
430 END
1000 REM  VERIFY BINGO CALL
1010 PRINT "[CLEAR]";:REM CLEAR WINDOW
1020 IF NP<4 THEN POSITION 1,5:PRINT #6;
"NOT ENOUGH NUMBERS":PRINT #6;"WERE CALL
ED YET":GOTO 370
1030 FREEFLAG=0:REM INIT
1040 GRAPHICS 1
1050 PRINT #6;"ENTER BINGO NUMBERS"
1060 PRINT #6;"DON'T INCLUDE THE"
1070 PRINT #6;"PRECEDING LETTER"
1080 PRINT #6
1090 PRINT #6;"ENTER f FOR FREE"
1100 PRINT #6;"ENTER l FOR LIST"
1110 PRINT #6
1120 FOR LOOP=1 TO 5
```

```
1130 PICK(LOOP)=0
1140 NEXT LOOP
1150 PRINT "ENTER THE WINNING NUMBERS"
1160 PRINT "(DON'T INCLUDE THE PRECEDING
 LETTER"
1170 GOSUB 15000:REM INPUT
1180 IF XX$="L" THEN GOSUB 2000:PRINT "E
NTER WINNING NUMBERS"
1190 IF XX$="F" THEN IF FREEFLAG=0 THEN
FREEFLAG=1:RT=RT+1:GOTO 1290
1200 IF XX=0 THEN PRINT "ENTER JUST THE
DIGITS - ENTER F FOR   FREE (BUT ONLY ON
CE)":GOTO 1170
1210 PICK(RT)=XX:IF RT<1 THEN 1250:REM S
KIP CHECK FOR REPEATS
1220 FOR LOOP=0 TO RT-1
1230 IF XX=PICK(LOOP) THEN PRINT "THAT W
AS ALREADY ENTERED- AGAIN?":PICK(LOOP)=0
:POP :GOTO 1170
1240 NEXT LOOP
1250 FOR LOOP=1 TO NP
1260 IF N(LOOP)=XX THEN RT=RT+1:POP :GOT
O 1290
1270 NEXT LOOP
1280 PRINT "THAT NUMBER IS NOT PICKED YE
T":GOSUB 14000:GOTO 370
1290 IF RT=5 THEN GOTO 3000
1300 GOTO 1170
2000 REM  PRINT ALL CALLED NUMBERS
2010 GRAPHICS 1:ROW=0:COL=1:REM INIT
2020 FOR LOOP=1 TO NP
2030 NM=N(LOOP)
2040 POSITION COL,ROW
2050 ROW=ROW+1:IF ROW>18 THEN ROW=0:COL=
COL+4
2060 GOSUB 4000
2070 NEXT LOOP
2080 RETURN
3000 REM  SOMEONE WON
3010 GRAPHICS 2
3020 FOR X=1 TO 3
3030 PRINT #6;"** a winner **"
3040 NEXT X
3050 PRINT #6
3060 PRINT #6;"ANOTHER GAME?";
```

```
3070 GOSUB 15100:REM GET
3080 IF XX$="Y" THEN RUN
3090 END
4000 REM  PRINT THE BINGO NUMBER WITH
 PRECEDING LETTER
4010 LT=INT((NM-1)/15+1)
4020 PRINT #6;BINGO$(LT,LT);NM
4030 RETURN
7000 REM  INIT AND FILL ARRAY
7010 FOR X=0 TO 75
7020 N(X)=X
7030 NEXT X
7040 RETURN
7100 REM SHUFFLE THE ARRAY OF NUMBERS
7110 FOR X=1 TO 75
7120 X1=INT(RND(1)*75)+1
7130 X2=N(X)
7140 N(X)=N(X1)
7150 N(X1)=X2
7160 NEXT X
7170 RETURN
10000 REM  PRINT BINGO CARD
10010 GOSUB 10900:REM SHUFFLE
10020 LINE$="  +------+------+------+---
---+------+"
10030 SPACE$="  |      |      |      |
     |      |"
10040 PART$="  |  "
10050 LPRINT :LPRINT "     OFFICIAL ATAR
I BINGO CARD"
10060 LPRINT :LPRINT
10070 LPRINT "     B      I      N
G      O"
10080 FOR LINE=1 TO 5
10090 B$=STR$(N(LINE)):IF LEN(B$)<2 THEN
 B$(LEN(B$)+1)=" "
10100 I$=STR$(N(LINE+15))
10110 N$=STR$(N(LINE+30)):IF LINE=3 THEN
 N$="XX"
10120 G$=STR$(N(LINE+45))
10130 O$=STR$(N(LINE+60))
10140 LPRINT LINE$
10150 LPRINT SPACE$
10160 LPRINT PART$;B$;PART$;I$;PART$;N$;
```

```
PART$;G$;PART$;O$;"   |"
10170 LPRINT SPACE$
10180 NEXT LINE
10190 LPRINT LINE$
10200 FOR LOOP=1 TO 7:LPRINT :NEXT LOOP
10210 RETURN
10900 REM SHUFFLE FOR CARD
10910 FOR LOOP=1 TO 61 STEP 15
10920 FOR LOOP2=LOOP TO LOOP+14
10930 TEMP=INT(RND(1)*15)+LOOP
10940 TEMP2=N(LOOP2)
10950 N(LOOP2)=N(TEMP)
10960 N(TEMP)=TEMP2
10970 NEXT LOOP2
10980 NEXT LOOP
10999 RETURN
14000 REM  PAUSE
14001 REM PAUSE IS DEFAULT AT 999
14002 REM TO SET YOUR OWN PAUSE
14003 REM SIMPLY DO THIS ON THE CALL
14004 REM
14005 REM PAUSE=300 : GOSUB 14020
14006 REM
14007 REM NOTE GOSUB AT 14020 IF YOU
14008 REM SET YOUR OWN PAUSE LENGTH
14009 REM
14010 PAUSE=999
14020 FOR LOOP=1 TO PAUSE
14030 NEXT LOOP
14099 RETURN
15000 REM  INPUT SUBROUTINE
15010 PRINT "[BELL]";
15020 POKE 764,255
15030 TRAP 15030
15040 INPUT XX$
15050 TRAP 15070
15060 XX=VAL(XX$)
15070 TRAP 34567:RETURN
15100 REM  'GET' SUBROUTINE
15110 PRINT "[BELL]";
15120 POKE 764,255
15130 TRAP 15130
15140 CLOSE #4
15150 OPEN #4,4,0,"K:"
15160 GET #4,XX
```

```
15170 CLOSE #4
15180 XX$=CHR$(XX)
15190 TRAP 15220
15200 XX=0
15210 XX=VAL(XX$)
15220 TRAP 34567:RETURN
```

```
10 REM <<< PHANTOM >>>
20 REM LEN LINDSAY,1980
30 DIM LETTER(10),XX$(3),FOUND(9),NUMBER
(3),GUESS$(3),CLASS$(20)
40 REM FOLLOWING ARE VARIABLES USED FOR
TARGET LINE NUMBERS
50 INSTRUCTIONS=1000:HINT=3000:DONE=9000
:EXPLAIN=2000:TRANSLATE=8000:REPLY=7000
60 CHECK=6000:ASK=270:RATING=4000:WINNER
=5000:RIGHT=5500:WRONG=5700
70 HINTCOUNT=0:GUESSCOUNT=0:FOUNDCOUNT=0
:WRONGCOUNT=0:REM INIT
80 GRAPHICS 1:SETCOLOR 2,0,0
90 FOUND(LOOP)=0:REM INIT ARRAY
100 FOR LOOP=0 TO 9:LET LETTER(LOOP)=ASC
("A")+LOOP:FOUND(LOOP)=0:NEXT LOOP:REM I
NIT ARRAYS
110 LET LETTER(10)=-1:REM INIT FOR INCOR
RECT INPUT CHECK
120 POSITION 2,5:PRINT #6;"phantom numbe
rs"
130 FOR LOOP=0 TO 9:REM SHUFFLE
140 MOVE=INT(RND(1)*10)
150 TEMP=LETTER(MOVE)
160 LET LETTER(MOVE)=LETTER(LOOP)
170 LET LETTER(LOOP)=TEMP
180 NEXT LOOP
190 GUESSCOUNT=1:REM INIT
200 PRINT "[CLEAR]At any time you may ty
pe HELP (for"
210 PRINT "instructions), HINT (to revea
l one"
220 PRINT "digit), or END (to stop)."
230 PRINT "Do you need instructions?";
240 GOSUB 15000
```

```
250 PRINT #6;"[CLEAR]";
260 IF XX$(1,1)="Y" THEN GOTO INSTRUCTIO
NS
270 PRINT "[CLEAR]";
280 IF HINTCOUNT+FOUNDCOUNT>9 THEN GOTO
DONE
290 FOR LOOP=65 TO 74
300 IF LOOP=70 THEN PRINT :PRINT " ";
310 RRINT CHR$(LOOP);"=";
320 FOR LOOP2=0 TO 9
330 IF LETTER(LOOP2)=LOOP THEN IF FOUND(
LOOP2)>0 THEN PRINT LOOP2;:POP :GOTO 360
340 NEXT LOOP2
350 PRINT "?";
360 PRINT " ";
370 NEXT LOOP
380 PRINT
390 PRINT "WHAT DO YOU THINK?"
400 GOSUB 15000:REM INPUT ROUTINE
410 GUESSCOUNT=GUESSCOUNT+1:REM ERRONEOU
S GUESSES ARE SUBTRACTED BY EXPLAIN
420 IF LEN(XX$)<>3 THEN GOTO EXPLAIN
430 GUESS$=XX$:REM XX$ FROM INPUT ROUTIN
E IS ASSIGNED TO GUESS$
440 IF GUESS$="HEL" THEN GOTO INSTRUCTIO
NS
450 IF GUESS$="HIN" THEN GOTO HINT
460 IF GUESS$="END" THEN GOTO DONE
470 IF GUESS$(1,1)>"J" OR GUESS$(1,1)<"A
" THEN GOTO EXPLAIN
480 GOSUB TRANSLATE
490 IF GUESS$(3,3)>"J" OR GUESS$(3,3)<"A
" THEN GOTO CHECK:REM CHECK FOR =
500 REM *** GOSUB TRANSLATE
510 IF GUESS$(2,2)="+" THEN ANSWER=NUMBE
R(1)+NUMBER(3):GOTO REPLY
520 IF GUESS$(2,2)="-" THEN ANSWER=NUMBE
R(1)-NUMBER(3):GOTO REPLY
530 IF GUESS$(2,2)="*" THEN ANSWER=NUMBE
R(1)*NUMBER(3):GOTO REPLY
540 GOTO EXPLAIN
550 IF GUESS$(2,2)<>"=" THEN GOTO EXPLAI
N
1000 PRINT "[CLEAR]";:REM INSTRUCTIONS
```

```
1010 PRINT "The ten digits (0123456789)
have"
1020 PRINT "each been replaced by letter
s"
1030 PRINT "(ABCDEFGHIJ). START=Continue
";:GOSUB 2920
1040 PRINT "[CLEAR]Each letter is used o
nly once, and"
1050 PRINT "all digits have been assigne
d one"
1060 PRINT "letter as its disguise. STAR
T=More";:GOSUB 2920
1070 PRINT "[CLEAR]You must figure out w
hat digit each"
1080 PRINT "letter represents. You are a
llowed to"
1090 PRINT "ask simple arithmetic expres
sions"
1100 PRINT "such as:  A+C   or  D-F. STA
RT=More";:GOSUB 2920
1110 PRINT "[CLEAR]I will tell you the l
etter that"
1120 PRINT "represents the last digit of
 the "
1130 PRINT "answer to your problem."
1140 GOSUB 2900
1150 PRINT "[CLEAR]I will not tell you i
f the answer has"
1160 PRINT "more than one digit, nor if
it is"
1170 PRINT "positive or negative. Good l
uck!!!"
1180 GOSUB 2900
1999 REM GOTO EXPLAIN  WHICH FOLLOWS IMM
EDIATELY
2000 REM EXPLAIN
2010 GUESSCOUNT=GUESSCOUNT-1:REM ERROR S
O SUBTRACT A GUESS TO BE FAIR
2020 PRINT "[CLEAR]";:REM CLEAR SCREEN
2030 PRINT "To guess the identity of a l
etter you"
2040 PRINT "type the letter, followed by
 an equal"
2050 PRINT "sign, then the digit that yo
u think"
```

```
2060 PRINT "is its disguised digit.  STA
RT=More";:GOSUB 2920
2070 PRINT "[CLEAR]Type  A=3 RETURN"
2080 PRINT "        ||[Z][R][R]>the NUMBER
"
2090 PRINT "        |[Z][R][R]>an EQUAL SI
GN"
2100 PRINT "        [Z][R][R]>the LETTER
   START=More";
2110 GOSUB 2920
2120 PRINT "[CLEAR]If you can't guess a
letters identity"
2130 PRINT "you can submit a simple addi
tion,"
2140 PRINT "subtraction, or multiplicati
on prob:"
2150 PRINT "such as D*F  or   J-A    STA
RT=More";:GOSUB 2920
2160 PRINT "[CLEAR]type A-C RETURN"
2170 PRINT "        ||[Z][R][R]>any LETTER
(A-J)"
2180 PRINT "        |[Z][R][R]>OPERATION S
IGN(+-*)"
2190 PRINT "        [Z][R][R]>any LETTER(A
-J) START=More";:GOSUB 2920
2199 GOTO ASK
2900 REM HIT START TO CONTINUE ROUTINE
2910 PRINT "HIT START TO CONTINUE";
2920 IF PEEK(53279)<>6 THEN 2920
2930 IF PEEK(53279)=6 THEN 2930:REM WAIT
 TILL LET UP ON START KEY
2999 RETURN
3000 REM HINT
3010 HINTCOUNT=HINTCOUNT+1
3020 GOSUB 7900:REM POSITION CURSOR FOR
PRINT#6
3030 FOR LOOP=9 TO 0 STEP -1
3040 IF FOUND(LOOP)=0 THEN FOUND(LOOP)=1
:PRINT #6;"-->";CHR$(LETTER(LOOP)+32);"=
";LOOP;:POP :GOTO ASK
3050 NEXT LOOP
3060 PRINT "NO HINT":REM SHOULD NEVER EX
ECUTE THIS LINE OR THE NEXT LINE
```

```
3099 GOTO ASK
4000 REM RATING
4010 SCORE=1000
4020 IF HINTCOUNT=0 THEN 4060
4030 FOR LOOP=1 TO HINTCOUNT
4040 SCORE=SCORE/2
4050 NEXT LOOP
4060 SCORE=SCORE-WRONGCOUNT*10
4070 SCORE=SCORE-GUESSCOUNT*2
4080 SCORE=INT(SCORE)
4090 IF SCORE<0 THEN SCORE=10:REM MINIMU
M SCORE
4099 RETURN
5000 REM WINNER
5010 PRINT "YOU WIN. ALL IDENTITIES ARE
UNCOVERED"
5020 GOTO DONE
5500 REM RIGHT GUESS
5510 PRINT "[BELL]";:REM BEEP
5520 GOSUB 7900:REM POSITION CURSOR FOR
PRINT#6
5530 PRINT #6;GUESS$;"yes";
5599 RETURN
5700 REM WRONG GUESS
5710 WRONGCOUNT=WRONGCOUNT+1
5720 GOSUB 7900:REM POSITION CURSOR FOR
PRINT#6
5730 PRINT #6;GUESS$;" no";
5799 RETURN
6000 REM CHECK IF = IS RIGHT
6010 IF GUESS$(2,2)<>"=" THEN GOTO EXPLA
IN
6020 TRAP EXPLAIN:REM TRAP NON NUMERIC
6030 NUMBER(3)=VAL(GUESS$(3,3))
6040 TRAP 34567:REM RELEASE TRAP
6050 IF ASC(GUESS$(1,1))<>LETTER(NUMBER(
3)) THEN GOSUB WRONG:GOTO ASK
6100 IF FOUND(NUMBER(3))>0 THEN GOSUB 79
00:PRINT #6;GUESS$;"---";:GOTO ASK:REM A
LREADY FOUND IT
6200 GOSUB RIGHT
6300 FOUNDCOUNT=FOUNDCOUNT+1
6310 FOUND(NUMBER(3))=2
6400 IF FOUNDCOUNT+HINTCOUNT>9 THEN GOTO WINNER
```

```
6500 GOTO ASK
7000 REM REPLY
7010 ANSWER=ABS(ANSWER)
7020 ANSWER=INT(ANSWER)
7030 IF ANSWER>9 THEN ANSWER=ANSWER-INT(
ANSWER/10)*10:GOTO 7010
7100 GOSUB 7900:REM SET CURSOR FOR PRINT
#6
7110 IF ROW=20 THEN ROW=0:COL=COL+6:IF C
OL>15 THEN PRINT #6,"[CLEAR]";:ROW=0:COL
=1
7120 POSITION COL,ROW
7130 PRINT #6;GUESS$;"=";CHR$(LETTER(ANS
WER))
7300 GOTO ASK
7900 ROW=ROW+1
7910 IF ROW=20 THEN ROW=0:COL=COL+6:IF C
OL>15 THEN PRINT #6,"[CLEAR]";:ROW=0:COL
=1
7920 POSITION COL,ROW
7999 RETURN
8000 REM TRANSLATE
8010 NUMBER(1)=10:NUMBER(3)=10:REM INIT
8020 FOR LOOP=1 TO 3 STEP 2
8030 FOR TEST=0 TO 9
8040 IF CHR$(LETTER(TEST))=GUESS$(LOOP,L
OOP) THEN NUMBER(LOOP)=TEST
8050 NEXT TEST
8060 NEXT LOOP
8099 RETURN
9000 REM DONE
9010 GRAPHICS 0
9020 PRINT "[CLEAR]";:REM CLEAR SCREEN
9030 PRINT "THANK YOU FOR PLAYING"
9040 PRINT
9050 GUESSCOUNT=GUESSCOUNT-1
9060 PRINT "WITH ONLY ";GUESSCOUNT;" GUE
SSES"
9070 PRINT "         AND ";WRONGCOUNT;" WRO
NG GUESS";:IF WRONGCOUNT<>1 THEN PRINT "
ES";
9080 PRINT
9090 PRINT "         AND ";HINTCOUNT;" HINT
";:IF HINTCOUNT<>1 THEN PRINT "S";
```

```
9100 PRINT
9110 IF FOUNDCOUNT=0 THEN PRINT "YOU COU
LD DEDUCE NONE OF THE NUMBERS":GOTO 9250
9120 PRINT "YOU FOUND THE IDENTITY OF "
9130 IF FOUNDCOUNT=10 THEN PRINT "ALL OF
 THE NUMBERS":GOTO 9150
9140 PRINT FOUNDCOUNT;" OF THE NUMBERS."
9150 PRINT
9160 GOSUB 14000
9170 FOR LOOP=0 TO 9
9180 PRINT LOOP;" WAS REPLACED BY ";CHR$
(LETTER(LOOP));
9190 IF FOUND(LOOP)>1 THEN PRINT " (YOU
FOUND IT)";
9200 PRINT
9210 NEXT LOOP
9220 PRINT
9230 GOSUB RATING
9240 PRINT "YOUR RATING IS: ";SCORE
9250 RESTORE 9500+SCORE/10
9260 READ CLASS$
9270 PRINT "YOUR CLASS IS:  ";CLASS$
9280 PRINT
9290 PRINT "HIT START FOR ANOTHER CASE";
9510 DATA BEGINNER
9520 DATA ROOKIE
9535 DATA PRIVATE
9540 DATA CHIEF
9545 DATA INSPECTOR
9550 IF PEEK(53279)<>6 THEN 9550
9560 RUN
9570 DATA CHIEF INSPECTOR
9580 DATA DETECTIVE
9586 DATA CHIEF DETECTIVE
9590 DATA SLEUTH
9594 DATA CHIEF SLEUTH
9597 DATA SUPER SLEUTH
9600 DATA SUPER SLEUTH***
9999 END
14000 REM WINNING SOUNDS
14010 FOR L1=40 TO 0 STEP -8
14020 FOR L2=40 TO L1 STEP -1:SOUND 0,L2
,10,10:NEXT L2
14030 NEXT L1:SOUND 0,0,0,0
14099 RETURN
```

```
15000 REM INPUT SUBROUTINE WITH BEEP
15001 REM XX$ IS STRING TO BE INPUT
15002 REM XX IS THE NUMERIC VALUE OF THE
 INPUT
15003 REM MUST DIM XX$(N)  -  N IS THE L
IMIT YOU WISH ON THE INPUT STRING LENGTH
15010 PRINT "[BELL]";:REM BUZZER
15020 POKE 764,255:REM CLEAR BUFFER
15030 TRAP 15030:REM TRAP ERROR ON INPUT
15040 INPUT XX$:REM INPUT STRING
15050 TRAP 15090:REM TRAP IF NO NUMERIC
AVAILABLE IN STRING
15060 XX=VAL(XX$):REM NUMERIC
15090 TRAP 34567:REM SPRING TRAP
15099 RETURN
```

```
1 REM * <<< BRAINBUSTER (BRAINBOX)>>>
2 REM *      Copyright (C) 1980 by
3 REM *   Len Lindsay & Ted M. Kahn
5 GRAPHICS 2
10 GOSUB 20000
12 DIM A$(1),PZ(10)
100 MV=1
101 SETCOLOR 4,0,0
110 GOSUB 6000:REM DRAW BOARD
115 GOSUB 7000:REM RANDOM PIECES
116 GOSUB 7100:REM DRAW PIECES
118 GOSUB 30000:REM GIVE INSTRUCTIONS
120 TRAP 120
121 PRINT "[CLEAR]MOVE # ";MV;"?"
130 IF STRIG(JS)=0 THEN 130:REM WAIT TIL
L BUTTON LET UP
131 ST=STICK(JS):IF STRIG(JS)=1 THEN 131
138 GOSUB 140+(ST-5)
139 GOTO 155
140 M=3:RETURN
141 M=9:RETURN
142 M=6:RETURN
144 M=1:RETURN
145 M=7:RETURN
146 M=4:RETURN
148 M=2:RETURN
149 M=8:RETURN
150 M=5:RETURN
155 IF PZ(M)=0 THEN PRINT "[CLEAR]CHOOSE
 ONLY LIT BOXES - YOUR MOVE":GOTO 130
160 MV=MV+1
190 PZ(M)=0:REM TURN OFF CHOSEN BOX
191 GOSUB 7100:REM UPDATE BOARD
250 GOSUB 300+M*10
260 WN=0
265 GOSUB 7100
270 FOR Q=1 TO 9:WN=WN+PZ(Q):NEXT Q
280 IF WN=8 AND PZ(5)=0 THEN 7500:REM WI
NNER *****
290 IF WN=0 THEN 7600:REM LOSER **
300 GOTO 120:REM NEXT MOVE
310 PZ(2)=1-PZ(2)
311 PZ(4)=1-PZ(4)
312 PZ(5)=1-PZ(5)
```

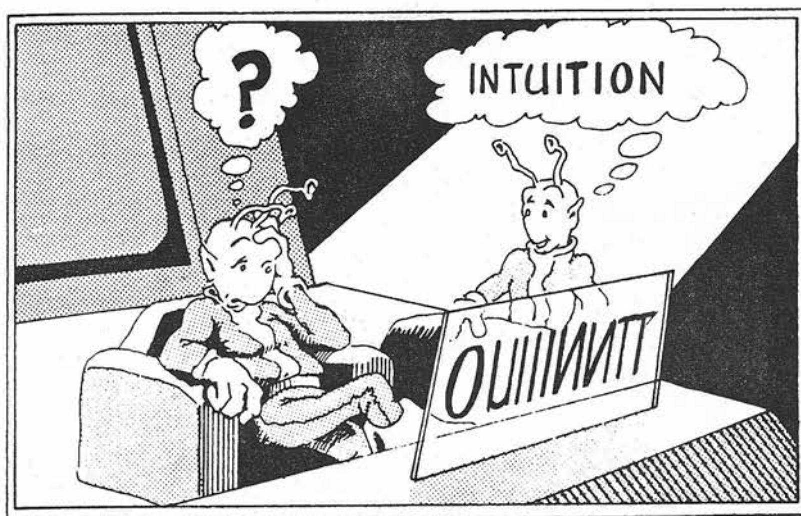
```
319 RETURN
320 PZ(1)=1-PZ(1)
321 PZ(3)=1-PZ(3)
329 RETURN
330 PZ(2)=1-PZ(2)
331 PZ(5)=1-PZ(5)
332 PZ(6)=1-PZ(6)
339 RETURN
340 PZ(7)=1-PZ(7)
341 PZ(1)=1-PZ(1)
349 RETURN
350 PZ(8)=1-PZ(8)
351 PZ(4)=1-PZ(4)
352 PZ(6)=1-PZ(6)
353 PZ(2)=1-PZ(2)
359 RETURN
360 PZ(9)=1-PZ(9)
361 PZ(3)=1-PZ(3)
369 RETURN
370 PZ(8)=1-PZ(8)
371 PZ(4)=1-PZ(4)
372 PZ(5)=1-PZ(5)
379 RETURN
380 PZ(7)=1-PZ(7)
381 PZ(9)=1-PZ(9)
389 RETURN
390 PZ(8)=1-PZ(8)
391 PZ(5)=1-PZ(5)
392 PZ(6)=1-PZ(6)
399 RETURN
999 END
1000 JS=99
1010 FOR X=0 TO 3
1020 IF STRIG(X)=0 THEN JS=X:GOSUB 1100
1030 NEXT X
1040 IF JS>3 THEN 1010
1050 RETURN
1100 IF STRIG(JS)=0 THEN 1100:REM WAIT T
ILL BUTTON RELEASED
1199 RETURN
6000 COLOR 3
6010 FOR Z=0 TO 12 STEP 6
6015 PLOT 10,Z:DRAWTO 28,Z
6020 FOR X=10 TO 28 STEP 6
```

```
6030 FOR Y=1 TO 5
6040 PLOT X,Y+Z
6050 NEXT Y
6060 NEXT X
6070 NEXT Z
6080 PLOT 10,18:DRAWTO 28,18
6099 RETURN
7000 FOR R=0 TO 10
7010 PZ(R)=INT(RND(0)+0.5)
7020 NEXT R
7099 RETURN
7100 PN=0
7110 FOR Z=12 TO 0 STEP -6
7120 FOR X=1 TO 13 STEP 6
7122 PN=PN+1:COLOR PZ(PN)
7125 FOR Y=1 TO 5
7130 PLOT X+10,Y+Z:DRAWTO X+14,Y+Z
7140 NEXT Y
7150 NEXT X
7160 NEXT Z
7190 RETURN
7500 GOSUB 8000:REM UPDATE CUMLATIVE
7570 PRINT "[CLEAR]YOU WON IN ";MV;" MOV
ES"
7571 SETCOLOR 4,4,4
7575 PRINT "PLAY AGAIN";
7576 INPUT A$
7590 IF A$="Y" THEN 100
7595 GRAPHICS 0:END
7600 GOSUB 8000:REM UPDATE
7605 PRINT "[CLEAR]YOU CAN'T WIN NOW"
7606 SETCOLOR 4,12,4
7610 GOTO 7575
8000 NG=NG+1
8010 AV=INT((AV*(NG-1)+MV)/NG)
8099 RETURN
20000 GRAPHICS 2:POSITION 4,1:PRINT #6;"
brain buster":POSITION 1,3:? #6;"COPYRIG
HT (C) 1980":POSITION 9,4:? #6;"BY"
20001 POSITION 0,6:? #6;"      LEN LINDSA
Y":POSITION 9,7:? #6;"AND":POSITION 6,8:
? #6;"TED KAHN"
20002 PRINT "THIS GAME REQUIRES A JOYSTI
CK.":? "TO START, PRESS THE RED FIRE BUT
```

```
TON"
20004 PRINT "ON YOUR JOYSTICK"
20005 JS=99
20010 FOR X=0 TO 3
20020 IF STRIG(X)=0 THEN JS=X
20030 NEXT X
20040 IF JS>3 THEN 20010
20050 GRAPHICS 3
20099 RETURN
30000 PRINT "[CLEAR]THE OBJECT OF THIS G
AME IS TO FINISH":? "WITH ALL BUT THE CE
NTER BOX LIT UP"
30005 PRINT "(PRESS THE RED BUTTON TO CO
NTINUE)...";:GOSUB 1000
30010 PRINT "[CLEAR]YOU SWITCH THE POSIT
IONS OF THE":? "BOXES BY MOVING YOUR JOY
STICK IN THE"
30020 PRINT "DIRECTION OF THE BOX YOU CH
OOSE AND":? "THEN PRESSING THE FIRE BUTT
ON...";:GOSUB 1000
30030 PRINT "[CLEAR]YOU MAY ONLY CHOOSE
BOXES WHICH ARE":? "ALREADY LIT.  CHOOSI
NG A LIT BOX "
30040 PRINT "WILL AFFECT THE STATE (LIT
OR NOT) OF":? "NEARBY BOXES...";:GOSUB 1
000
30050 PRINT "[CLEAR]REMEMBER:  MOVE THE
JOYSTICK UP TO ":? "CHOOSE TOP CENTER, D
OWN FOR BOTTOM"
30060 PRINT "CENTER; LEFT, RIGHT, OR DIA
GONALLY.";:GOSUB 1000
30070 PRINT "[CLEAR]WHEN YOU HAVE MADE Y
OUR CHOICE (KEEP":? "ONE HAND ON THE JOY
STICK), THEN PRESS"
30080 PRINT "THE FIRE BUTTON...":GOSUB 1
000
30090 PRINT "[CLEAR]IF YOU TURN OFF A CO
RNER"
30100 PRINT "ITS THREE NEIGHBORS WILL CH
ANGE TOO":GOSUB 1000
30110 PRINT "[CLEAR]IF YOU TURN OFF THE
MIDDLE OF A SIDE"
```

```
30120 PRINT "THE TWO NEIGHBORS ON THAT S
IDE CHANGE":GOSUB 1000
30130 PRINT "[CLEAR]IF YOU TURN OFF THE
VERY CENTER BOX"
30140 PRINT "THE MIDDLE BOX ON EACH SIDE
 CHANGES":GOSUB 1000
30999 RETURN
```

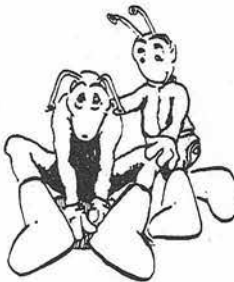
פרק ה'



משחקי מילים

באמצעות המחשב האישי של אטארי אפשר ליצור עשרות משחקי מלים משעשעים ובעלי אתגר. החל ממשחקים פשוטים של ניחוש אותיות ועד לתכניות מורכבות באסטרולוגיה שמספרות לאנשים על אישיותם ועתידם. אפשר לפתח תרגילים, לכתוב ספורים טפשיים וליצור משחקי חברה. הפרק הזה יציג בפניך את כל הסוגים השונים הללו של משחקי לשון. חשוב לזכור שמחשב אטארי מטפל במלים אחרת משהוא מטפל במספרים. אם שכחת מה שלמדת על הכרזות DIM ואת הדרך שבה נשמר מקום בזכרון עבור מלים, כדאי לחזור על הקטעים המתאימים בחלק הראשון, שמדברים על DIM, A$, A(X),ו- CLR.

נתחיל בתכנית פשוטה של ניחוש מלה-או אותיות, שאפשר לגוונה בתריסר דרכים שונות.



ניחוש מילה- משחק פשוט

תכנית זו, בת שש שורות, נותנת לאדם אחד לשמור מלה או אות בזכרון המחשב, מנקה מיד את המסך ואז נותנת הזדמנות למישהו אחר לנחש את המלה או את האות :

```
10 DIM A$(5),B$(5)
20 INPUT A$
30 PRINT "[CLEAR]"
40 INPUT B$
50 IF A$=B$ THEN PRINT "GOT IT":END
60 GOTO 30
```

שורה 10, מכריזה (DIM) על שני משתנים שמרשים למחשב לאחסן מלים באורך של עד חמש אותיות. כמובן, אפשר להכריז על A$ ועל B$ כעל משתנים בעלי יכולת אחסון של מלים יותר ארוכות, כרצונך. כדאי גם לשים לב, ששורה 30 מנקה את המסך. זה נעשה ע"י תקתוק מספר השורה ו-" PRINT

```
30 PRINT"
```

ואז לחיצה קודם על המקש ESCAPE ואז על SHIFT
ו CLEAR ביחד, ולבסוף מתקתקים. שוב " בסוף
השורה. כאשר מבצעים כך, הנה מה שצריך להופיע על
המסך :

```
30 PRINT "↑ "
```

והנה גרסה קצת שונה של משחק פשוט זה לשנים (או יותר)
שחקנים :



```
5 PRINT "PUT IN A 5 LETTER WORD FOR"
6 PRINT "A FRIEND TO GUESS"
10 DIM A$(5),B$(5)
20 INPUT A$
25 PRINT "[CLEAR]"
26 PRINT "TRY TO GUESS MY 5 LETTER WORD"
30 INPUT B$
40 IF A$=B$ THEN PRINT "GOOD YOU'VE GOT IT":END
50 GOTO 30
```

והנה גרסה שלישית, שבה נותן לך המחשב רמז, עד כמה
התקרבת בניחוש למלת המטרה. שורות 100 עד 120 אומרות
למחשב להדפיס כל אות שנחשת נכון במקום שבו היא
מופיעה במילת המטרה.

כך, אם המלה שצריך לנחש היא HEART
והניחוש היה START
המחשב ידפיס ART



```
5 REM BEGINNING OF WORD GUESSING GAME
10 DIM A$(5)
20 DIM B$(5)
30 PRINT "PICK A 5 LETTER WORD FOR YOUR FRIEND TO
   GUESS"
40 INPUT A$
50 PRINT "[CLEAR]"
60 PRINT "GUESS A 5 LETTER WORD"
70 INPUT B$
80 IF A$=B$ THEN PRINT "YOU GOT IT":END
90 PRINT "THESE LETTERS ARE IN YOUR WORD"
100 FOR X=1 TO 5
110 IF A$(X,X)=B$(X,X) THEN PRINT A$(X,X)
120 NEXT X
130 PRINT "TRY AGAIN"
140 GOTO 70
```

כמה שנויים יותר מורכבים בגרסה השלישית של
התכנית יפיקו רמז יותר מענין.

להוסיף:

```
25 DIM C$(5)
75 C$="00000"
105 R=INT(RND(1)*5)+1
```

לשנות:

```
110 IF C$(R,R)= "0" THEN C$(R,R)=A$(X,X): PRINT
   A$(R,R)
115 IF C$(R,R)< >"0" THEN 105
```

זה ידפיס את האותיות של המלה האמיתית בסדר אקראי (מבולבל), בלי קשר למה שתקתקת פנימה, (אלא אם כן קלעת למלה הנכונה).



התבוננות: הבזקי מילים

התכנית הזו הוצעה ע"י ד"ר דין בראון, חבר ותיק של אחד המחברים, המאמין שלמחשבים יכולת גדולה במתן מימדים חדשים לתפיסה ולחשיבה. תכנית מיוחדת זו מבוססת על העובדה הפשוטה, שהצגת מילה לזמן קצר ביותר גורמת למוח ליצור תמונה או דימוי. אם זה נעשה עם כמה מלים בזמן קצר, המתבונן יכול לקשר את הדימויים הללו וליצור אפילו דמויים יותר מורכבים.

הבחירה במלים בנות 4 אותיות (שורות 1000 עד 1002) לא היתה שרירותית :מלים בנות 4 אותיות מיוחדות ביכלתן ליצור את הדמויים הללו במוחנו.
ניתן להחליף את המלים שבחרנו במלים משלך, כל זמן שיש 10 מלים בשורה (שורה 10 גם מגבילה אותך למלים בנות 4 אותיות ; אפשר לשנות זאת, אם רוצים, ע"י שנוי האורך המותר בהכרזת משתנה המחרוזת($ WORD בשורה 30).

```
10 REM * < MEDITATIVE WORD IMAGERY >
20 REM *   suggested by Dean Brown
30 DIM WORD$(4)
40 GRAPHICS 1+16
50 POSITION 8,10
60 R=INT(RND(1)*2)+1000
70 RESTORE R
80 NOW=INT(RND(1)*10)+1
90 FOR I=1 TO NOW
100 READ WORD$
110 NEXT I
120 REM * FLASH WORD *
130 MAX=300+INT(RND(1)*600)+1:GOSUB 170
140 PRINT #6;WORD$
150 MAX=300:GOSUB 170
160 GOTO 40
170 REM PAUSE (WAIT) ROUTINE
180 FOR WW=1 TO MAX:NEXT WW
190 RETURN
200 END
1000 DATA WIND,RAIN,FIRE,SAND,SELF,FARM,
TREE,LAKE,ROCK,LAND
1001 DATA FOOD,FACE,SLUM,PLUM,BALL,SALT,
WALL,GAME,MATE,MIST
```



מילון ומשחקים למילים בנות שלוש אותיות

משחקי מלים רבים (כמו שבץ נא, תשבצים, וג'וטו) תלויים בשמוש במילון כלשהו כדי לקבוע רשימת מלים מותרות במשחק. אפשר להכין מילונים למשחקי המלים במחשב שלך ולשמור אותם על סרט או תקליטון.
בדרך זו, אחרי העבודה הראשונית של תקתוק המילון למחשב, יהיו בידך מילונים שונים לשימוש חוזר ונשנה במשחקים שונים. זהירות בענין אחד, כל כל פנים : צריך לדאוג שמספרי השורות במילון יהיו שונים ממספרי השורות בחלקים האחרים של תכניות המשחקים שלך.

כדי להתחיל, חברנו מילון באנגלית של מלים בנות שלש אותיות. התכנית שמאחסנת את המילון גם תבדוק אם צירוף של שלש אותיות הוא באמת מלה, ולכן תהיה שמושית מאד למגוון רחב של משחקי מלים. אנו מציעים לאחסן את השורות 16499-19000 כתת תכנית על סרט או תקליטון, ע"י שמוש בפקודה "LIST "C:, או ע"י פקודת "LIST "D: WORDS (או כל שם שנראה לך). אחר כך אפשר לאחד את התכנית עם המשחק המסוים שמעונינים בו, ע"י פקודת "ENTER "C: או "ENTER "D:WORDS.
יתכן שיהיה ברצונך לשמור גם את תת התכנית של הזהוי (שורות 0-320), כתת תכנית נפרדת או ביחד עם המילון.

```
10 REM * <<< 3 LETTER DICTIONARY >>>
20 REM *   bypassing string arrays
30 REM *      Copyright 1980 by
40 REM *          Len Lindsay
50 DIM T$(3),W$(3)
60 TRAP 60:PRINT "[CLEAR]ENTER A 3 LETTE
R WORD"
70 X=X+1:SETCOLOR 2,X,2
80 INPUT W$
90 IF LEN(W$)<>3 THEN 60
100 RESTORE 10000+100*ASC(W$)
110 FOR TEST=1 TO 45
120 READ T$
130 IF T$=W$ THEN POP :GOTO 160
140 NEXT TEST
150 GOTO 60
160 PRINT "CORRECT - TRY AGAIN?"
170 INPUT T$
180 IF T$(1,1)<>"N" THEN 60
16490 REM *  BEGINNING OF DATA FOR
16492 REM * THE 3 LETTER DICTIONARY
16494 REM *
16496 REM *     Add your own words
16498 REM * on the appropriate lines
16500 DATA ACE,ACT,ADD,ADO,ADZ,AFT,AGE,A
GO,AHA,AID,AIL,AIM,AIR,ALB,ALE,ALL,AND,A
NT,ANY,APE,ARC,ARE,ARK,ARM,ART,ASH
16510 DATA AUK,AWE,AWL,AXE,AYE
16600 DATA BAA,BAD,BAG,BAH,BAN,BAR,BAT,B
AY,BED,BEE,BEG,BET,BEY,BIB,BID,BIG,BIN,B
IT,BOA,BOB,BOO,BOP,BOW,BOX,BOY,BRA
```

```
16610 DATA BUD,BUG,BUM,BUN,BUR,BUS,BUT,B
UY
16700 DATA CAB,CAD,CAL,CAM,CAN,CAP,CAR,C
AT,CAW,CHI,COD,COG,CON,COO,COP,COT,COW,C
OY,CRY,CUB,CUD,CUE,CUM,CUP,CUR,CUT
16800 DATA DAB,DAD,DAM,DAW,DAY,DDT,DEN,D
EW,DID,DIE,DIG,DIM,DIN,DIP,DNA,DOE,DOG,D
ON,DOT,DRY,DUB,DUD,DUE,DUG,DUN,DUO
16810 DATA DYE
16900 DATA EAR,EAT,EBB,EEL,EFT,EGG,EGO,E
KE,EKG,ELF,ELK,ELL,ELM,EMU,END,EON,ERA,E
RE,ERG,ERR,ESP,ETA,EVE,EWE,EYE
17000 DATA FAD,FAG,FAN,FAR,FAT,FAY,FEE,F
EN,FEW,FEY,FEZ,FIB,FIE,FIG,FIN,FIR,FIT,F
IX,FLU,FLY,FOB,FOE,FOG,FOP,FOR,FOX
17010 DATA FRO,FRY,FUN,FUR
17100 DATA GAB,GAG,GAL,GAM,GAP,GAR,GAS,G
AT,GAY,GEE,GEL,GEM,GET,GIG,GIN,GNU,GOB,G
OD,GOO,GOT,GUM,GUN,GUT,GUY,GYP
17200 DATA HAD,HAG,HAH,HAM,HAT,HAW,HAY,H
EM,HEN,HER,HEW,HEX,HEY,HID,HIE,HIM,HIP,H
IS,HIT,HMM,HOB,HOD,HOE,HOG,HON,HOP
17210 DATA HOT,HOW,HUB,HUE,HUG,HUH,HUM,H
UN,HUT
17300 DATA ICE,ICY,ILK,ILL,IMP,INK,ION,I
OU,IRE,IRK,ITS,IVY
17400 DATA JAB,JAG,JAM,JAR,JAW,JAY,JET,J
EW,JIB,JIG,JOB,JOG,JOT,JOY,JUG,JUT
17500 DATA KEG,KEN,KEY,KID,KIN,KIP,KIT
17600 DATA LAB,LAC,LAD,LAG,LAM,LAP,LAW,L
AX,LAY,LED,LEE,LEG,LEI,LEK,LET,LEU,LID,L
IE,LIP,LIT,LOG,LOT,LOW,LOX,LSD,LUG
17610 DATA LYE
17700 DATA MAD,MAN,MAP,MAR,MAT,MAW,MAY,M
EN,MET,MEW,MID,MIL,MIN,MIX,MOA,MOB,MOD,M
OM,MOO,MOP,MOW,MUD,MUG,MUM
17800 DATA NAB,NAG,NAP,NAW,NAY,NEE,NET,N
EW,NIB,NIL,NIP,NIT,NIX,NOD,NOR,NOT,NOW,N
TH,NUB,NUT
17900 DATA OAF,OAK,OAR,OAT,OBI,ODD,ODE,O
FF,OFT,OHM,OIL,OLD,ONE,OOH,OPE,OPT,ORB,O
RE,OUR,OUT,OVA,OWE,OWL,OWN
18000 DATA PAD,PAL,PAN,PAP,PAR,PAT,PAW,P
AY,PEA,PEG,PEN,PEP,PER,PET,PHI,PIE,PIG,P
IN,PIP,PIT,PLY,POD,POI,POP,POT,POW
```

```
18010 DATA POX,PRO,PRY,PSI,PUB,PUG,PUN,P
UP,PUT
18200 DATA RAG,RAM,RAN,RAP,RAT,RAW,RAY,R
ED,REV,RHO,HIB,RID,RIG,RIM,RIP,RNA,ROB,R
OC,ROD,ROE,ROT,ROW,RUB,RUE,RUM,RUN
18210 DATA RUT,RYE
18300 DATA SAC,SAD,SAG,SAL,SAP,SAT,SAW,S
AX,SAY,SEA,SEC,SEE,SET,SEW,SEX,SHE,SHH,S
HY,SIC,SIN,SIP,SIR,SIT,SIX,SKI,SKY
18310 DATA SLY,SOB,SOD,SOL,SON,SOP,SOS,S
OT,SOW,SOX,SOY,SPA,SPY,STY,SUB,SUE,SUM,S
UN,SUP
18400 DATA TAB,TAG,GAI,TAN,TAP,TAR,TAT,T
AU,TAW,TAX,TEA,TEE,TEN,THE,TIC,TIE,TIN,T
IP,TIT,TOE,TOM,TON,TOO,TOP,TOT,TOW
18410 DATA TOY,TRY,TUB,TUG,TUN,TUX,TWI,T
WO
18500 DATA UGH,UHF,URN,USA,USE
18600 DATA VAN,VAT,VET,VEX,VHF,VIA,VIE,V
IP
18700 DATA WAD,WAG,WAN,WAR,WAS,WAX,WAY,W
EB,WED,WEE,WEN,WET,WHO,WHY,WIG,WIN,WIT,W
OE,WON,WOO,WOW
18900 DATA YAK,YAM,YAP,YAW,YEA,YEN,YES,Y
ET,YEW,YIP,YON,YOU
19000 DATA ZED,ZIP,ZOO,ZZZ
```

כיצד להשתמש במילון

מרגע שאחסנת את המילון הזה על סרט (ע"י פקודת "LIST "C:) או על תקליטון (בפקודת "LIST "D:NAME, כאשר NAME הוא שם המילון, עד 8 אותיות), אפשר להשתמש בו במספר רב של משחקי מלים. וזוהי הדרך לעשות זאת :

1. עליך לפתח את המשחק ולתקתק אותו במחשב, ולודא שמספר השורה האחרונה בתכנית אינו עולה על 9999.

2. השלב הבא הוא טעינת המילון מהסרט או מהתקליטון :
טעינה מסרט "ENTER "C:
טעינה מתקליטון "ENTER "D:NAME
זה יטען את המילון כך שהתכנית הקודמת לא תמחק מן הזכרון.

```
ENTER "C:" (from tape) or
ENTER "D:NAME" (from disk)
```

3.עכשו צריך לבדוק את התכנית שתקתקת כדי לראות אם אין בה שגיאות, ואם ברצונך - אפשר לשנות כמה דברים. ומעתה אפשר לאחסן את כל התכנית כיחידה שלמה אחת.

למשל, הנה תכנית פשוטה מאד שמדפיסה על המסך אילו מלים עומדות לרשותך במילון. כדאי לנסות אותה ולראות עד כמה קל הדבר "לחבר" משחקי מלים בדרך זו :

```
1 REM *** LIST OF ALL 3-LETTER WORDS *
2 REM *** AVAILABLE IN DICTIONARY ***
3 REM *** LEN LINDSAY, 2/28/80 ***
5 DIM W$(3)
10 POKE 82,0
20 PRINT "[CLEAR] HERE ARE THE THREE-
   LETTER WORDS I KNOW:"
30 READ W$
40 IF W$="ZZZ" THEN END
50 PRINT W$;" ";
60 GOTO 30
9999 REM *** 3-LETTER WORDS FOLLOW**
```

והנה עוד משחק מלים שמשתמש במילון. המשחק הזה מבוסס על המשחק המפורסם של לואיס קרול, "זוגיות" (DOUBLETS)。 "זוגיות" הם צורה של גלגולי מלים. מתחילים בשתי מלים, למשל RUN ו- SIT ומנסים לשנות את RUN ל- SIT。 החוקים לגלגול המלים הם :

1) אפשר לשנות רק אות אחת בכל פעם, 2) התוצאה של כל שנוי חייבת להיות מלה בעלת משמעות ו-3) אסור לשנות את סדר האותיות במילה - מוכרחים להחליף אות אחת באות אחרת. לפניך דרך אחת לשנות את המלה RUN למלה SIT בשלשה צעדים :

RUN
SUN
SIN
SIT

והנה תכנית לזוגיות בנות שלש אותיות, כאשר RUN ו- RED הן מילות המפתח ; נשתמש במילון כדי לקבוע אם מלים שקבלנו תוך כדי גלגול מילות המפתח מותרות או לא :

```
10 REM <<< SUNTORED >>>
20 REM LEN LINDSAY 1980
22 REM *       *** IMPORTANT ***
24 REM *  YOU MUST ADD THE 3 LETTER
26 REM *  DICTIONARY DATA STATEMENTS
28 REM *  STARTING AT LINE 16490
30 DIM S$(3),E$(3),W$(3),T$(3),Q$(3)
40 GRAPHICS 1
50 PRINT "DO YOU WANT TO USE YOUR OWN WO
RDS";
60 INPUT T$
70 IF T$(1,1)="Y" THEN GOSUB 9000:GOTO 1
00
80 S$="SUN":REM THE STARTING WORD
90 E$="RED":REM THE ENDING WORD
100 GRAPHICS 1
110 PRINT #6;"CHANGE ";S$;" TO ";E$
120 LINE=2:REM INIT START LINE TO PRINT
WORDS ON
130 W$=S$
140 POSITION 1,LINE
150 PRINT #6;W$
160 LINE=LINE+1
170 IF W$=E$ THEN 300:REM WIN
180 IF LINE>18 THEN PRINT #6;"YOU SEEM L
OST":GOTO 8000
190 PRINT "WHAT WORD COMES NEXT?"
200 INPUT T$
210 RESTORE 10000+100*ASC(T$)
220 FOR TEST=1 TO 45
230 READ Q$
240 IF Q$="ZZZ" THEN POP :GOTO 270
250 IF Q$=T$ THEN POP :GOTO 400
260 NEXT TEST
270 PRINT "I DON'T KNOW THAT WORD - ANOT
HER?"
299 GOTO 200
300 FOR L1=40 TO 0 STEP -8
310 FOR L2=40 TO L1 STEP -1:SOUND 0,L2,1
0,10:NEXT L2
320 NEXT L1:SOUND 0,0,0,0
330 POSITION 0,19:PRINT #6;"WINNER WITH
";LINE-3;" WORDS"
399 GOTO 8000
```

```
400 MATCH=0
410 FOR TEST=1 TO 3
420 IF T$(TEST,TEST)=W$(TEST,TEST) THEN
MATCH=MATCH+1
430 NEXT TEST
440 IF MATCH=2 THEN W$=T$:GOTO 140
499 PRINT "EXACTLY 1 LETTER MUST CHANGE"
:GOTO 200
8000 PRINT "WOULD YOU LIKE TO TRY AGAIN"
;
8010 INPUT T$
8020 IF T$(1,1)="N" THEN END
8030 PRINT "WITH THE SAME WORDS";
8040 INPUT T$
8050 IF T$(1,1)="Y" THEN GOTO 100
8060 PRINT "DO YOU WANT TO USE YOUR OWN
WORDS";
8070 INPUT T$
8080 IF T$(1,1)="Y" THEN GOSUB 9000:GOTO
 100
8099 GOTO 100
8199 GOTO 100
9000 PRINT "[CLEAR]"
9010 PRINT "WHAT IS THE STARTING WORD";
9020 INPUT T$
9030 GOSUB 9500
9040 IF T$="***" THEN PRINT "ILLEGAL WOR
D - TRY AGAIN":GOTO 9010
9050 S$=T$
9110 PRINT "WHAT IS THE ENDING WORD";
9120 INPUT T$
9130 GOSUB 9500
9140 IF T$="***" THEN PRINT "ILLEGAL WOR
D - TRY AGAIN":GOTO 9110
9150 IF S$=T$ THEN PRINT "ENDING WORD CA
N'T BE THE SAME":GOTO 9110
9160 E$=T$
9200 RETURN
9500 RESTORE 10000+100*ASC(T$)
9520 FOR TEST=1 TO 45
9550 READ Q$
9560 IF Q$="ZZZ" THEN POP :T$="***":GOTO
 9599
9570 IF Q$=T$ THEN POP :GOTO 9599
9580 NEXT TEST
```

```
9590 T$="***"
9599 RETURN
10000 REM *  ADD 3 LETTER DICTIONARY
10010 REM *  DATA STATEMENTS TO THIS
10020 REM *  PART OF THE PROGRAM  -
10030 REM *  LINES 16490 TO 19000
```

נסו להפגש עם חברים שיש ברשותם מחשב אישי של אטארי, ולחבר מילון משלכם למלים בנות ארבע או חמש אותיות. כמובן, ככל שהמלים ארוכות יותר, כך יהיו משחקי המלים מורכבים ומעניינים יותר.

הנה כמה רעיונות למשחקים שאפשר לתכנת אותם בקלות, תוך שמוש במילון למלים בנות שלש אותיות :

1. משחק שמתחיל באות אחת,שהיא עיצור. מטרת המשחק היא לראות מי יכול ליצור את המספר הגדול ביותר של מלים שמתחילות באות הזו. מטרה אחרת יכולה להיות לראות מי ייצור את מספר המלים בנות שלש אותיות שיכילו את האות הזו בתוכן.

2. אפשר לתכנן משחק דומה, שיהיה מבוסס על תנועות. (5 אותיות התנועה באנגלית הן a e i o u). יש לבחור בתנועה, ולראות כמה מלים בנות שלש אותיות יכולים ליצור, שהתנועה תופיע בהן, בכל מקום שהוא במלה. ינצח מי שיכתוב את מספר המלים הרב ביותר.

3. בוחרים שתי מלים בנות שלש אותיות מתוך רשימת המלים במילון, באופן אקראי. כמה מלים אחרות בנות שלש אותיות אפשר ליצור משתי המלים האלה ? אילו יכולת לבחור כרצונך מתוך המלים, אילו מלים היו המתאימות ביותר כדי ליצור את המספר הגדול ביותר של מלים ?

4. אם נותנים לך 5 (או כל מספר אחר גדול מארבע) אותיות, כאשר אחת מהן לפחות היא תנועה, כמה מלים בנות שלש אותיות אפשר ליצור מהן ? משחק זה אפשר לשחק ביחידות, או כתחרות בין כמה שחקנים. יש כאן גם אתגר תכנותי : לבנות לוח תוצאות אל תוך המשחק, כך שכל שחקן יקבל נקודה לזכותו עבור כל מלה נכונה.

הערה של לן לינדסי, לגבי זהוי מלים בנות שלש אותיות.
היא מכוונת למתכנתים מנוסים:

דרכי זיהוי למילים בנות שלוש אותיות במחשב אטארי

"הנה תכנית מהירה אך "נקיה" למשתמשי אטארי.
ודאי כבר ידוע לך שמחשבי אטארי אינם מרשים מערכים של מחרוזות, אבל המגבלה הזו אינה פוסלת משחקי מלים וכד'.

אטארי מאפשר לשחזר DATA מכל מספר שורה, גם שורת משתנה שבפקוח התכנה. אני משתמש בתכונה זו כדי להמנע ממערכי מלים. במקום זאת אני משתמש במשפטי DATA כמערך לשעת הצורך.

אני מסתמך על השמוש ב- ASC של לוח המקשים שב- ATARI BASIC, שיאמר לי את קוד התוים ATASCII של האות הראשונה בכל מחרוזת.

כדי לזהות את כל המלים בנות שלש אותיות, צריך שתהיה למחשב רשימה מלאה של המלים. אני השתמשתי ברשימה של DYMAX של כל המלים החוקיות בנות שלש אותיות, מתוך המילון

AMERICAN HERITAGE ELEMENTARY SCHOOL DICTIONARY

אלה מופיעות בשורות ה- DATA 16500-19000.

יש עובדה נוספת שהתכנית שלי משתמשת בה, המתירה לא יותר משניה אחת לזהוי מלה נכונה. יש 45 מלים שמתחילות באות S, יותר מאשר בכל אות אחרת. כך, הלולאה המתחילה בשורה 200 צריכה לבדוק 45 ערכים לכל היותר, וזה יכול להעשות בפחות משניה.

התכנית הבאה תבקש ממך לתקתק כל מלה שהיא.
אח"כ תאמר לך אם שלש האותיות הראשונות של המלה שתקתקת מהוות מלה חוקית בת שלש אותיות. פשוט מאד, אך בסיס טוב להרחבה בעתיד".

```
0 REM (C) 1980 LEN LINDSAY
10 DIM W$(3), T$(3) : REM THIS IS ALL THE SPACE WE WILL
   NEED TO RESERVE
100 TRAP 100 : PRINT "[CLEAR] ENTER A
   3 LETTER WORD" : REM SET OUR TRAP
105 X=X+1 : SETCOLOR 2,X,2 : REM CHANGE THE BACK-
   GROUND COLOR
110 INPUT W$
120 IF LEN(W$) < >3 THEN 100 : REM NOT A THREE LETTER
   WORD
130 RESTORE 10000+100*ASC(W$) : REM RESTORE DATA AT
   FIRST LETTERS LINE
200 FOR TEST = 1 TO 45 : REM MAXIMUM NUMBER OF
   WORDS TO CHECK
210 READ T$
220 IF T$=W$ THEN POP : GOTO 300 : REM CORRECT & POP
   OUT OF LOOP
230 IF T$="ZZZ" THEN 100 : REM END OF WORDS AND NO
   MATCH
240 NEXT TEST
250 GOTO 100 : REM NO MATCH FOUND
300 PRINT "CORRECT - TRY AGAIN?"
310 INPUT T$
320 IF T$< >"N" THEN 100 : REM ANSWER "N" TO STOP
16499 REM DATA STATEMENTS FOLLOW
```

סדרות

זהו משחק שבו המטרה היא לקשר מלים זו לזו תוך שמוש באות הראשונה והאחרונה שלהן. על שחקן אחד לתקתק מלה בת שלש אותיות והשחקן השני צריך להמשיך במלה שהאות הראשונה שלה זהה לאות ה<u>אחרונה</u> של המלה הקודמת. למשל, הנה רצף אפייני:

PEN
NOT
TIN
וכו'

צריך לזכור לטעון (בפקודת ENTER) את המילון של המלים בנות שלש אותיות, אחרי שמתקתקים את התכנית ולפני שמתחילים לשחק.

```
10 REM <<<  SERIES  >>>
20 REM LEN LINDSAY 1980
22 REM *       *** IMPORTANT ***
24 REM *  YOU MUST ADD THE 3 LETTER
26 REM *  DICTIONARY DATA STATEMENTS
28 REM *  STARTING AT LINE 16490
30 NUMBER=554
40 DIM PICKED(NUMBER),PLAYER1$(8),PLAYER
2$(8)
50 DIM IN$(15),LASTW$(14),STARTL$(1)
60 DIM TEST$(14),S(1),P(1),T(1)
70 GAMEOVER=9000
80 OPTION=3010:GOSUB OPTION-10
90 GOSUB 2000:REM GRAPHICS & COLORS
100 PRINT #6;"3 LETTER WORD SERIES"
110 POSITION 6,8:PRINT #6;"TO BEGIN"
120 POSITION 5,9:PRINT #6;"HIT  start"
130 POSITION 2,2:PRINT #6;"FOR INSTRUCTI
ONS"
140 POSITION 5,3:PRINT #6;"HIT option"
150 POSITION 6,11:PRINT #6;" ATARI "
160 KEY=PEEK(53279)
170 IF KEY=3 THEN GOSUB 1000:GOTO 90
180 IF KEY=5 THEN GOSUB OPTION:OPTION=OP
TION+10:GOSUB 4020:IF OPTION>3030 THEN O
PTION=3000
190 IF KEY=6 THEN 250
200 POSITION 0,5:PRINT #6;PLAYER1$;" VS
";PLAYER2$;"HIT select TO CHANGE"
210 SETCOLOR 3,0,RND(1)*256
220 FOR DELAY=1 TO 50:NEXT DELAY
230 SETCOLOR 2,0,PN:PN=PN+2
240 GOTO 160
250 GRAPHICS 2+16:PRINT #6;"WAIT"
260 FOR A=1 TO NUMBER:PICKED(A)=0:NEXT A
:REM INIT
270 S(0)=0:S(1)=0
280 T(0)=0:T(1)=0
290 COUNT=0
300 PLAYER=1
310 TEST$=PLAYER2$:GOSUB 5000:PLAYER2$=T
EST$
320 PASS=0
330 PLAYER=1-PLAYER
```

```
340 POSITION 0,10:GOSUB 6900
350 POSITION 0,11:GOSUB 6900
360 IF PASS>1 THEN GOTO GAMEOVER
370 COUNT=COUNT+1
380 GOSUB 6000
390 POSITION 1,3:PRINT #6;"LAST WORD WAS
:"
400 POSITION 1,4:GOSUB 6900:REM ERASE LI
NE
410 IF LEN(LASTW$)>1 THEN IF LASTW$(1,1)
<="Z" THEN TEST$=LASTW$:GOSUB 5000:POSIT
ION 1,4:PRINT #6;TEST$
420 POSITION 1,6:PRINT #6;"NEXT WORD MUS
T"
430 POSITION 3,7:PRINT #6;"START WITH: "
;
440 STARTL$=CHR$(INT(RND(1)*26)+65)
450 IF LEN(LASTW$)>1 THEN STARTL$=LASTW$
(LEN(LASTW$))
460 PRINT #6;CHR$(ASC(STARTL$)+160)
470 POSITION 1,8
480 IF P(PLAYER)=0 THEN PRINT #6;"ENTER 
YOUR WORD"
490 IF P(PLAYER)=1 THEN PRINT #6;"MY WOR
D IS:     "
500 GOSUB 6900
510 IF P(PLAYER)=1 THEN GOSUB 7000:GOTO 
670:REM COMPUTER TURN
520 CHAR=1
530 IN$="              "
540 POSITION 1,9:PRINT #6;IN$
550 OPEN #1,4,0,"K"
560 POKE 764,255:REM CLEAR BUFFER
570 IF PEEK(764)=255 THEN SETCOLOR 3,0,R
ND(1)*256:GOTO 570
580 GET #1,IN
590 CLOSE #1
600 IF IN=126 AND CHAR>1 THEN CHAR=CHAR-
1:IN$(CHAR,CHAR)=" ":GOTO 540:REM DELETE
 HIT
610 IF IN=155 THEN 670:REM RETURN HIT
620 IF IN<65 AND IN<>32 OR IN>90 THEN 55
0:REM NOT ALPHABETIC
630 IN$(CHAR,CHAR)=CHR$(IN)
640 CHAR=CHAR+1
```

```
650 IF CHAR>15 THEN CHAR=15
660 GOTO 540
670 IF LEN(IN$)<1 THEN 500
680 IF CHAR>1 THEN IN$=IN$(1,CHAR-1):REM
 MINUS TRAILING SPACES
690 IF IN$="PASS" THEN PASS=PASS+1:GOTO
330
700 IF ASC(IN$)<>ASC(STARTL$) THEN PASS=
PASS+1:PRINT #6;"1st letter's not ";STAR
TL$:GOTO 900
710 RESTORE
720 FOR TEST=1 TO NUMBER
730 READ TEST$
740 IF TEST$=IN$ THEN POP :GOTO 800
745 IF TEST$(1,1)>IN$(1,1) THEN TEST=NUM
BER
750 NEXT TEST
760 PRINT #6;"not a legal word"
770 GOTO 900:REM CONTINUE
800 IF PICKED(TEST)>0 THEN PRINT #6;"tha
t word is used   hit start":GOSUB 4000:G
OSUB 6000:GOTO 340
810 PICKED(TEST)=PLAYER+1
820 PRINT #6;"that word works"
830 LASTW$=IN$
840 S(PLAYER)=S(PLAYER)+1
850 T(PLAYER)=T(PLAYER)+LEN(IN$)
900 PRINT #6;"HIT start";
910 GOSUB 6050
920 GOSUB 4000:IF KEY<>6 THEN 920
930 GOTO 320
999 END
1000 GRAPHICS 1+16:GOSUB 2010
1010 PRINT #6;"THREE LETTER WORDS"
1020 PRINT #6;"ARE THE LEGAL WORDS"
1030 PRINT #6;"THE LAST LETTER OF"
1040 PRINT #6;"THE PREVIOUS WORD"
1050 PRINT #6;"MUST BE THE FIRST"
1060 PRINT #6;"LETTER OF THE NEXT"
1070 PRINT #6;"WORD. NO REPEATS"
1075 PRINT #6
1080 PRINT #6;"type 'PASS' if you"
1090 PRINT #6;"can not move."
1095 PRINT #6
1100 PRINT #6;"AN ILLEGAL MOVE IS"
```

```
1110 PRINT #6;"TREATED AS PASSING"
1120 PRINT #6;"YOUR TURN"
1125 PRINT #6
1130 PRINT #6;"THE GAME IS OVER"
1140 PRINT #6;"IF BOTH PLAYERS PASS"
1200 PRINT #6
1900 PRINT #6;"HIT start"
1990 GOSUB 4000:IF KEY<>6 THEN 1990
1999 RETURN
2000 GRAPHICS 2+16
2010 SETCOLOR 0,2,8:REM CAPS - REG TEXT
2020 SETCOLOR 1,12,10:REM lower LUMIN -
OPTIONS IN STARTER
2030 SETCOLOR 2,8,12:REM CAPS WINDOW - T
ITLE IN STARTER
2040 SETCOLOR 3,4,6:REM lower - NAMES OF
 THE SPECIAL KEYS - VIBRATES NAMES
2050 SETCOLOR 4,0,0:REM BACKGROUND
2099 RETURN
3000 PLAYER1$="human   ":PLAYER2$="compu
ter":P(0)=0:P(1)=1
3009 RETURN
3010 PLAYER1$="computer":P(0)=1
3019 RETURN
3020 PLAYER2$="human   ":P(1)=0
3029 RETURN
3030 PLAYER1$="human   ":P(0)=0
3039 RETURN
4000 IF PEEK(53279)<>7 THEN 4000
4010 KEY=PEEK(53279):IF KEY=7 THEN SETCO
LOR 3,0,RND(1)*255:GOTO 4010
4020 IF PEEK(53279)<>7 THEN 4020
4099 GOSUB 2010:RETURN
5000 FOR A=1 TO LEN(TEST$)
5010 TEST$(A,A)=CHR$(ASC(TEST$(A,A))+128
)
5020 NEXT A
5099 RETURN
6000 GOSUB 2010
6010 TEST$=PLAYER1$:GOSUB 5000:PLAYER1$=
TEST$
6020 TEST$=PLAYER2$:GOSUB 5000:PLAYER2$=
TEST$
6030 POSITION 0,0
```

```
6040 PRINT #6;PLAYER1$;" VS ";PLAYER2$
6050 POSITION 1,1:PRINT #6;S(0);":";T(0)
6060 POSITION 13,1:PRINT #6;S(1);":";T(1
)
6900 PRINT #6;"                    ";
6999 RETURN
7000 RESTORE
7010 FOR TEST=1 TO NUMBER
7020 SETCOLOR 3,3,RND(1)*256
7030 READ IN$
7040 IF ASC(IN$)=ASC(STARTL$) AND PICKED
(TEST)=0 THEN POP :GOTO 7100
7050 NEXT TEST
7060 IN$="PASS"
7070 POSITION 1,10
7080 PRINT #6;"I PASS! HIT start"
7090 GOSUB 4000:IF KEY<>6 THEN 7090
7100 CHAR=LEN(IN$)+1
7110 POSITION 1,9:PRINT #6;IN$
7199 RETURN
9000 POSITION 1,10
9010 PRINT #6;"[CLEAR]"
9020 FOR LL=20 TO 0 STEP -10
9030 FOR L=50 TO LL STEP -1.5:SOUND 0,L,
10,10:NEXT L:SOUND 0,0,0,0
9040 NEXT LL
9050 GOSUB 6000
9060 PRINT #6
9070 POSITION 3,10:PRINT #6;"GAME IS OVE
R"
9080 POSITION 4,11:PRINT #6;"HIT start";
9090 GOSUB 4000:IF KEY<>6 THEN 9090
9099 RUN
10000 REM *  ADD 3 LETTER DICTIONARY
10010 REM *  DATA STATEMENTS TO THIS
10020 REM *  PART OF THE PROGRAM  -
10030 REM *  LINES 16490 TO 19000
```



צופן סודי

בכל רחבי העולם משתמשים במחשבים לפענוח צפנים. מחשב אטארי שלך יוכל לשמש אותך ליצירת צופן, לכתיבת הודעות מוצפנות ולפענוחן.

הצופן הפשוט ביותר מבוסס על החלפת אות אחת באחרת. הנה תכנית למשחק שתאפשר לך לתכנן החלפת אותיות כזאת. אחרי תקתוק הצופן, המחשב יבחן אותך ע"י הבזקת אות ויבקש ממך את הצופן עבור האות הזאת. זה כשלעצמו מהוה שיטה טובה לזכור צופן שיצרת בעצמך:

```
5 REM *** CODE83 ALPHABET CODE GUESSING
  GAME
6 REM (C) BY TED M. KAHN, 1980
10 DIM A$(26),B$(26),G$(1),X$(1)
20 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
30 PRINT "[CLEAR] INPUT YOUR SECRET CODE FOR EACH
   PRINT "LETTER OF THE ALPHABET. USE EACH"
32 PRINT "LETTER ONLY ONCE!"
40 FOR I=1 TO 26
45 PRINT "THE CODE FOR ";A$(I,I);" = ";
50 INPUT X$:IF I=1 THEN GOTO 58
52 FOR J=1 TO (I-1)
54 IF B$(J, J)=X$ THEN POP:PRINT "YOU ALREADY USED
   THAT ONE":GOTO 45
56 NEXT J
58 B$(I, I)=X$
60 NEXT I
70 REM *** NOW FOR THE GUESSING GAME*
80 PRINT "[CLEAR]":REM *** CLEAR SCREEN***
90 LET R=INT(RND(1)*26)+1
100 PRINT "WHAT'S THE CODE FOR ";A$(R,R);
110 INPUT G$
120 IF G$=B$(R,R) THEN PRINT "GREAT! YOU GOT IT!":
    GOTO 90
130 PRINT "NO, THE CODE IS";B$(R,R)
140 GOTO 90
```

דרך אגב, אם כל אות תעמוד בפני עצמה, תהיה לך תכנית שתוכל לעזור לילד או ילדה שרק מתחילים ללמוד לקרוא אנגלית, לשנן את האותיות.

מחשב אטארי שלך מכיל אותיות וכן סימנים גרפיים. את הסמנים הגרפיים מקבלים ע"י לחיצה על מקש (CTRL)

ובאותו זמן על אחת האותיות. יש טבלה של כל הסמנים
הגרפיים במדריך למשתמש של אטארי לשפת BASIC.
הנה תכנית צופן שבה מיוצגת כל אות ע"י הסימן הגרפי
שמופיע במקש של אותה אות. למשל, (H) יוצפן ע"י ◢
ו (T) ע"י ● . התכנית תבחן אותך על מיקום הסימנים
הגרפיים בלוח המקשים.

```
10 REM *   <<< GRAPHICS CODES >>>
20 REM * For learning which keys
30 REM * correspond to which control
40 REM * graphics characters
50 REM * (C) by Ted M. Kahn, 1980
60 DIM A$(26),B$(26),G$(1)
70 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
80 REM * LINE 90 CONTAINS [CTRL] GRAPHIC
S CODES OF EACH LETTER OF THE ALPHABET,
IN ORDER
90 B$="[A][B][C][D][E][F][G][H][I][J][K]
[L][M][N][O][P][Q][R][S][T][U][V][W][X][
Y][Z]"
100 REM * NOW FOR THE GUESSING GAME*
110 PRINT "[CLEAR]"
120 LET R=INT(RND(1)*26)+1
130 PRINT :PRINT "WHAT LETTER GOES WITH
";B$(R,R);
140 INPUT G$
150 IF G$=A$(R,R) THEN PRINT "GREAT! YOU
 GOT IT!":GOTO 120
160 PRINT "NO, THE LETTER FOR ";B$(R,R);
" IS ";A$(R,R)
170 GOTO 120
```

והנה תכנית מלאה להצפנה ופענוח. באמצעותה אפשר להכניס
את הצופן ולקבל את ההודעה המוצפנת. אפשר גם לתקתק פנימה
הודעה מוצפנת, והמחשב יפענח אותה עבורך. (אם ברצונך
לקבל הדפסה של ההודעות במדפסת - אטארי 820, אטארי 822
או אטארי 825 - עליך לכתוב LPRINT במקום PRINT בשורות
450, 470, 630 ו-640). לא לשכוח גם את הרווח בסוף
המחרוזת ALPHA$ בשורה 80. הוא חשוב ביותר!

```
10 REM * <<< ENCODING & DECODING >>>
20 REM *  (C) 1981 by Ted M. Kahn
30 REM * Alphabetic codes only.  All
40 REM * other characters will remain
50 REM * as is
```

```
60 REM
70 DIM ALPHA$(27),CODE$(27),MESSAGE$(100
),CMESSAGE$(100),A$(1)
80 LET ALPHA$="ABCDEFGHIJKLMNOPQRSTUVWXY
Z "
90 REM
100 REM *  ENCODING
110 REM
120 PRINT "[CLEAR]TYPE IN YOUR ENCODING
SYSTEM,":PRINT "ONE LETTER PER LINE."
130 PRINT "(FOR EXAMPLE:  'A=E' MEANS 'A
 WILL BE REPLACED BY E')":PRINT
140 FOR I=1 TO 26
150 PRINT ALPHA$(I,I);"=";
160 INPUT A$:IF I=1 THEN 210
170 FOR J=1 TO I-1
180 IF CODE$(J,J)<>A$ THEN 200
190 PRINT "YOU'VE ALREADY USED THAT LETT
ER":PRINT :POP :GOTO 150
200 NEXT J
210 CODE$(I,I)=A$
220 NEXT I
230 CODE$(27,27)=" "
240 PRINT :PRINT "DO YOU WANT TO ENCODE
OR DECODE":PRINT "A MESSAGE";
250 INPUT A$
260 IF A$="E" THEN GOSUB 290
270 IF A$="D" THEN GOSUB 490
280 GOTO 240
290 REM
300 REM *  ENCODE A MESSAGE
310 REM
320 PRINT "[CLEAR]"
330 MESSAGE$="":CMESSAGE$=""
340 PRINT "TYPE IN THE MESSAGE YOU WANT
TO ENCODE"
350 PRINT "(UP TO A MAXIMUM OF 100 CHARA
CTERS)":PRINT
360 INPUT MESSAGE$
370 L=LEN(MESSAGE$)
380 FOR I=1 TO L
390 FOR J=1 TO 27
400 IF MESSAGE$(I,I)<>ALPHA$(J,J) THEN G
OTO 420
```

```
410 CMESSAGE$(I,I)=CODE$(J,J):POP :GOTO
440
420 NEXT J
430 IF J>27 THEN CMESSAGE$(I,I)=MESSAGE$
(I,I)
440 NEXT I
450 PRINT "YOUR ENCODED MESSAGE IS:"
460 REM * REPLACE 'PRINT' IN LINES 450 &
 470 WITH 'LPRINT' FOR PRINTOUT
470 PRINT CMESSAGE$
480 RETURN
490 REM
500 REM *  DECODE A MESSAGE
510 REM
520 CMESSAGE$="":MESSAGE$=""
530 PRINT :PRINT "TYPE IN THE ENCODED ME
SSAGE":PRINT
540 INPUT CMESSAGE$
550 L=LEN(CMESSAGE$)
560 FOR I=1 TO L
570 FOR J=1 TO 27
580 IF CMESSAGE$(I,I)<>CODE$(J,J) THEN G
OTO 600
590 MESSAGE$(I,I)=ALPHA$(J,J):POP :GOTO
620
600 NEXT J
610 IF J>27 THEN MESSAGE$(I,I)=CMESSAGE$
(I,I)
620 NEXT I
630 PRINT "[CLEAR]YOUR DECODED MESSAGE I
S:"
640 PRINT MESSAGE$:RETURN
650 REM * REPLACE 'PRINT' IN 630 & 640
660 REM * WITH 'LPRINT' FOR PRINTOUT
```

אותיות משוכלות

שיכול אותיות היא דרך להצפין מלים ע"י ערבוב האותיות.
UNS היא מלה משוכלת (אנגרמה) של SUN ו-MSANAGRA של המלה ANAGRAMS.
אנו מביאים כמה תכניות שאחרי "קישוט" יתקבלו מהן משחקי שיכול משוכללים.

בתכנית הראשונה משכלים מלה, מכניסים אותה לזכרון המחשב ומבקשים ממישהו אחר לנחש את המלה המקורית.

```
10 REM *     <<< ANAGRAM 1 >>>
20 REM *  (C) 1981 by Ted M. Kahn
30 REM * longest word is 30 letters
40 DIM WORD$(30),SWITCH$(30),ANAGRAM$(30)
50 ANAGRAM$=""
60 PRINT :PRINT "TYPE IN THE WORD YOU WI
SH TO ANAGRAM"
70 INPUT WORD$:L=LEN(WORD$)
80 GOSUB 120
90 PRINT "THE ANAGRAM IS:"
100 PRINT ANAGRAM$
110 GOTO 50
120 REM *  ANAGRAM SUBROUTINE
130 FOR I=1 TO L
140 SWITCH$(I,I)="0"
150 NEXT I
160 FOR J=1 TO L
170 X=INT(RND(1)*L)+1
180 IF SWITCH$(X,X)="1" THEN GOTO 170
190 SWITCH$(X,X)="1"
200 ANAGRAM$(X,X)=WORD$(J,J)
210 NEXT J
220 IF ANAGRAM$=WORD$ THEN GOTO 120
230 RETURN
```

התכנית השניה מספקת שגרה, או תת תכנית כללית, המערבבת ומבלבלת את האותיות במלה. המלה מסופקת על ידך, והמחשב מבלבל אותה.

```
10 REM GENERAL ANAGRAM PROGRAM
15 PRINT "[CLEAR]"
20 DIM WORD$(40),ANA$(40),COUNT(40)
30 PRINT "TYPE IN YOUR WORD";
35 TRAP 30
40 INPUT WORD$:ANA$="":L=LEN(WORD$)
42 REM CLEAR LETTER-REPEAT COUNTER
45 FOR X=1 TO L:COUNT(X)=0:NEXT X
50 FOR I=1 TO L:REM SCRAMBLE LETTERS
60 R=INT(RND(0)*L)+1:REM SHIFT LETTER
70 IF COUNT(R)=1 THEN 60
```

```
75 COUNT(R)=1
80 ANAS(R,R)=WORDS(I,I)
90 NEXT I
92 REM VERIFY ANAGRAM
95 IF WORDS=ANAS THEN ANAS="":GOTO 45
100 PRINT "THE ANAGRAM IS: ";ANAS
105 PRINT
110 GOTO 30
```

לבסוף הנה תכנית שמספקת מלים משוכלות ומשחק שהוא אתגר. המשחק בנוי לשני שחקנים;אחת (או אחד) שבוחרת את המלה שאותה צריך לנחש,ושניה (או שני) שמנחשת אותה מן המלה המשוכלת.

```
10 REM * << ANAGRAM GUESSING GAME >>
20 REM *   (C) 1981 by Ted M. Kahn
30 REM * longest word is 30 letters
40 DIM WORD$(30),SWITCH$(30),ANAGRAM$(30
),GUESS$(30)
50 GRAPHICS 0
60 ANAGRAM$=""
70 PRINT :PRINT "TYPE IN THE WORD YOU WA
NT YOUR FRIEND":PRINT "TO GUESS:"
80 INPUT WORD$:L=LEN(WORD$)
90 GOSUB 240
100 GRAPHICS 2
110 POSITION 5,4
120 PRINT "OK.  HERE'S MY WORD.  TRY TO
GUESS IT":PRINT "(YOU HAVE 10 GUESSES)"
130 PRINT #6;ANAGRAM$:PRINT
140 FOR G=1 TO 10
150 PRINT "GUESS ";G;
160 INPUT GUESS$
170 IF GUESS$=WORD$ THEN GOSUB 350:POP :
GRAPHICS 0:GOTO 60
180 IF G<>10 THEN PRINT "NO, TRY AGAIN"
190 NEXT G
200 PRINT "THE ANSWER IS ";WORD$
210 FOR WAIT=1 TO 1000:NEXT WAIT
```

```
220 GRAPHICS 0
230 GOTO 60
240 REM *  ANAGRAM SUBROUTINE
250 FOR I=1 TO L
260 SWITCH$(I,I)="0"
270 NEXT I
280 FOR J=1 TO L
290 X=INT(RND(1)*L)+1
300 IF SWITCH$(X,X)="1" THEN GOTO 290
310 SWITCH$(X,X)="1"
320 ANAGRAM$(X,X)=WORD$(J,J)
330 NEXT J:IF ANAGRAM$=WORD$ THEN 240
340 RETURN
350 REM *  YOU GOT IT
360 GRAPHICS 2:POSITION 2,4
370 PRINT #6;"YOU GOT IT!"
380 FOR WAIT=1 TO 500:NEXT WAIT:RETURN
```



סיפורים בשברים

ספורים בשברים הם ספורים כתובים רק בחלקם, שאפשר להשלימם ולהפכם לספורי סבתא, או למעשיות שלא היו ולא נבראו. זו דוגמה של ספור שבור, שאפשר להשלימו :

יום אחד יצא __________ העצוב __________ ארוך.
למחרת יצא __________ __________. השנים נפגשו
ו __________.

תכנית פשוטה שתתאים כמעט לכל ספור שבור, יכולה להתחיל בצורה הבאה :

```
10 REM A SIMPLE FRACTURED PHRASE
20 DIM A$(25), B$(25)
30 PRINT "ONE DAY A";
40 INPUT A$
50 PRINT "WENT TO THE ";
60 INPUT B$
70 PRINT "HERE IS YOUR SENTENCE:"
80 PRINT "ONE DAY A ";A$;" WENT TO THE ";B$
```

(ונא לזכור את הרווחים בשורה 80).

בשורה 30 יש רווח בין A והמרכאות הסוגרות, ובשורה 50 יש רווח בין המרכאות הפותחות ובין W, ורווח נוסף בין E והמרכאות הסוגרות.

ע"י התאמת גודל המחרוזת המותר עפ"י הכרזת DIM, תהיה לך האפשרות לכתוב את הספורים השבורים שלך.
הנה עוד ספור שבור ומסובך שתעבוג לשחק אותו עם חברים או במסיבות

```
10 REM * <<< SLIGHTLY FRACTURED >>>
20 REM * <<<     FAIRY TALES     >>>
30 REM * developed by Robert A. Kahn
40 REM * modified by Ted M. Kahn
50 DIM Y$(20),F$(20),R$(20),J$(20)
60 DIM B$(20),L$(20),A$(20),V$(20)
70 PRINT "[CLEAR]":REM CLEAR THE SCREEN
80 PRINT " A SLIGHTLY FRACTURED FAIRY TA
LE":PRINT :PRINT
90 PRINT "WHAT'S YOUR NAME";
100 INPUT Y$:PRINT
110 PRINT "AND THE NAME OF"
120 PRINT "YOUR BEST FRIEND";
130 INPUT F$:PRINT
140 PRINT "THE NAME OF YOUR"
150 PRINT "FAVORITE RELATIVE";
160 INPUT R$:PRINT
170 PRINT "JOB TITLE OF A KIND OF PERSON
"
180 PRINT "WHO ALWAYS BUGS YOU";
190 INPUT J$:PRINT
200 PRINT "THE NAME OF YOUR BOSS";
210 INPUT B$:PRINT
220 PRINT
230 PRINT "JUST A FEW MORE QUESTIONS...
"
240 PRINT
250 PRINT "NAME FOR A TYPE OF"
260 PRINT "LARGE BUILDING";
270 INPUT L$:PRINT
280 PRINT "ONE WORD FOR WHAT YOU WOULD L
IKE TO DO";
290 PRINT "WHEN YOU'RE ANGRY";
300 INPUT A$:PRINT
310 PRINT "NAME OF A LUXURY VEHICLE YOU'VE"
```

```
320 PRINT "ALWAYS WANTED TO OWN";
330 INPUT V$
340 PRINT "[CLEAR]"
350 PRINT "          THE THREE LITTLE PIGS
"
360 PRINT
370 PRINT "ONCE THERE WERE THREE LITTLE
PIGS"
380 PRINT "NAMED ";Y$;", ";F$;" AND ";R$
;"."
390 PRINT
400 PRINT "A HUNGRY ";J$;" NAMED ";B$
410 PRINT "WANTED TO EAT THEM FOR SUPPER
."
420 PRINT
430 PRINT "TERRIFIED, THEY HID IN THEIR
";L$;"."
440 PRINT "FURIOUS, ";B$;" SHOUTED: "
450 PRINT "'I'LL HUFF, AND I'LL PUFF AND
 I'LL"
460 PRINT A$;" YOUR ";L$;" DOWN!'"
470 PRINT
480 PRINT "NO FOOLS, THE THREE PIGS SNEA
KED OUT "
490 PRINT "THE BACK DOOR AND ESCAPED IN
THEIR"
500 PRINT V$;"."
510 PRINT
520 PRINT "              T H E  E N D"
530 END
```

המכתב השבור הזא וריאציה של הספור השבור. זוהי תכנית למכתב שבור לנציב תלונות הצבור. (גירסה זו של המשחק מניחה שיש ברשותך מדפסת כלשהי של אטארי. אם אין לך, עלית לשנות את כל פקודות ה LPRINT ל PRINT.

```
10 REM * <<< A TYPICAL FORM LETTER >>>
20 REM * PRINTS LETTER ON YOUR PRINTER
30 REM *  by Robert Kahn & Lee Berman
40 REM * slightly modified by Ted Kahn
50 DIM D$(20),N$(30),A$(35),C$(20),S$(10
),Z$(5),M$(10),G$(40),P$(40)
60 PRINT "[CLEAR]BE SURE YOUR LINE PRINT
```

```
ER IS ON!":PRINT
70 PRINT "INPUT DATA FOR FORM LETTER:"
80 PRINT :PRINT
90 PRINT "DATE OF LETTER";
100 INPUT D$:PRINT :PRINT
110 PRINT "NAME OF CITIZEN";
120 INPUT N$:PRINT
130 PRINT "STREET ADDRESS";
140 INPUT A$:PRINT
150 PRINT "CITY";
160 INPUT C$:PRINT
170 PRINT "STATE";
180 INPUT S$:PRINT
190 PRINT "ZIP CODE";
200 INPUT Z$:PRINT :PRINT
210 PRINT "CONTRIBUTION";
220 INPUT M$:PRINT
230 PRINT
240 PRINT "YOUR GROUP'S NAME";
250 INPUT G$:PRINT
260 PRINT
270 PRINT "PROBLEM";
280 INPUT P$:PRINT
290 REM ARK -- LETTER GENERATED HERE
300 PRINT "[CLEAR]":PRINT "TOP MARGIN?";
:INPUT S:FOR I=1 TO S:LPRINT :NEXT I
310 LPRINT
320 LPRINT "                OFFICE OF
     "
330 LPRINT "           THE HONORABLE SENAT
OR  "
340 LPRINT "               RALPH WINDBAG
     "
350 LPRINT :LPRINT
360 LPRINT "                          ";D$
370 LPRINT :LPRINT
380 LPRINT N$
390 LPRINT A$
400 LPRINT C$;", ";S$;" ";Z$
410 LPRINT
420 LPRINT "DEAR ";N$;":"
430 LPRINT
440 LPRINT "I'D LIKE TO THANK YOU, PERSO
NALLY,"
450 LPRINT "FOR YOUR GENEROUS CONTRIBUTI
```

```
ON OF"
460 LPRINT M$;".  GROUPS LIKE YOURS,"
470 LPRINT G$;", WHO ARE"
480 LPRINT "WILLING TO STAND UP AND BE C
OUNTED,"
490 LPRINT "HAVE MADE IT POSSIBLE FOR US
 TO"
500 LPRINT "TRIUMPH IN OUR AGGRESSIVE CA
MPAIGN "
510 LPRINT "AGAINST ";P$;"!"
520 LPRINT
530 LPRINT "MANY PEOPLE ARE WORRIED THAT
 NO ONE"
540 LPRINT "HERE IN THE STATE CAPITAL HA
S TIME"
550 LPRINT "FOR SMALL GROUPS LIKE YOURS.
"
560 LPRINT "BUT I WANT TO REASSURE YOU,
"
570 LPRINT "PERSONALLY, THAT ";G$;" WILL
 ALWAYS"
580 LPRINT "BE MY PRIME CONCERN WHILE IN
 OFFICE."
590 LPRINT
600 LPRINT "              SINCERELY YOURS,
"
610 LPRINT "              SENATOR RALPH WI
NDBAG"
```

גירסה אחרונה של ספורים שבורים עוסקת בספורים שחוברו בידי המחשב. המתכנתת (או המתכנת) מתקתקת חלק מן הספור, והמחשב בוחר מלים באופן אקראי מתוך מילון שמתכנתים פנימה כדי להשלים את הספור. הטכניקות שתארנו בקטע על משחקי מלים בנות שלש אותיות ואלה שהבאנו כאן, מקלות על פתוח תכניות לכתיבת עלילות מדע בדיוני בידי המחשב, או חיבור "כותרות" המפרסמות את הסרטים האחרונים במדע בדיוני. התכנית שתייצר עבורך את הפנינים הללו איננה קשה לכתיבה. באמצעותה, יהיה בידך מילון של חלקי ה"כותרות" האהובים עליך, באותו אופן שהוצג הדבר במילון למשחקי מלים.

לפנינו כמה דוגמאות של "הכרזות אימים", שניתן לצפות להן מתכנית מסוג כזה. בדוגמאות הללו, שנלקחו מתכנית

שפותחה במרכז המדעי LAWRENCE HALL OF SCIENCE באוניברסיטת קליפורניה בברקלי, המלים באותיות גדולות הן אלה שמחבר המחשב. המלים באותיות הקטנות הן מלים שמתקתק המשתמש במחשב.

סיפור א'

EARTH SCIENTISTS DISCOVER GIANT prehistoric birds WHICH (WHO) fly under the influence of alcohol AND ARE usually vegetarian AND . . . CANNOT BE KILLED BY toothpicks, SO THEY EAT US ALL UP (YUMMY)!!
THE END

סיפור ב'

EARTH SCIENTISTS DISCOVER GIANT snow peas WHICH (WHO) LOOK UPON US AS A SOURCE OF NOURISHMENT AND. . .CANNOT BE KILLED BY pleasant meditating giraffes, SO SCIENTISTS INVENT A WEAPON THAT TURNS THEM INTO peanut butter.
THE END

לתכניות האלה שתי צורות, בדרך כלל : האחת, שבה עושה המחשב את כל העבודה ורק בוחר באופן אקראי חלקים שונים ומסדר אותם יחד והשניה, שבה מספק האדם חלקים מן הספור ובאלה משתמש המחשב יותר מאוחר (לעתים קרובות בזמן בלתי צפוי).

אנו משאירים לך לכתוב את התכנית הזו לבד. רמז אחד הוא שפקודת ON/GOSUB S1,S2,S3,S4 יכולה להיות מאד שמושית בבנית תכנית כזאת. ורמז נוסף : אפשר ליצור מחרוזות תוים נפרדות לחלקי משפט שונים : למשל,

(פועל) VERB$ ="EATS CRUSHES TRAMPLES FLIES MELTS";

(נושא) SUBJECT$ = "FLYING-SAUCERS MIDDLE-AGED RHINOCEROCES GARANTUAN -THREE-LEGGED- LIZARDS

וכו'. ע"י שמוש במערך מספרי מקביל, אפשר לעקוב אחרי תחילתו של כל "חלק" ע"י ציון מספר התו הראשון בתוך המחרוזת. אז אפשר להשתמש במספר אקראי כדי לבחור איזה מבין כמה נושאים, פעלים וכו' ישמש לספור שיכתוב המחשב.

תכניות מסוג זה מתוכננות לבעלי דמיון פרוע, אבל יכולות לשמש גם ליצירת תאורים של מראות יפים ושקטים.
הכל תלוי בחומר הגלם שניתן למחשב.



מגיד עתידות

מי לא רצה פעם שיהיה לו מגיד עתידות פרטי משלו ? בתוכנית הזו, נוכל להפוך את מחשב אטארי למגיד עתידות... במובן מסוים. אבל, בואו נחשוב רגע : זה הרי מחשב, נכון ? והרי יש אנשים שאומרים שהמחשב אף פעם לא טועה !

```
10 REM *   <<< FORTUNE TELLER >>>
20 REM *      copyright (C) 1980
30 REM *  by Kohl, Kahn & Lindsay
40 DIM NAME1$(20),NAME2$(20),NAME3$(20),
NAME4$(20),TEXT$(20)
50 PRINT "[CLEAR]"
60 PRINT "I AM THE ORACLE!"
70 PRINT "I WILL TELL YOU WHO YOU SHALL
MARRY"
80 PRINT "IF YOU GIVE ME 4 POSSIBLE NAME
S"
90 PRINT
100 GOSUB 240:NAME1$=TEXT$
110 GOSUB 240:NAME2$=TEXT$
120 GOSUB 240:NAME3$=TEXT$
130 GOSUB 240:NAME4$=TEXT$
140 PRINT "[CLEAR]"
150 PRINT "I'M LOOKING INTO MY CRYSTAL B
ALL..."
160 PRINT :FOR WW=1 TO 500:NEXT WW
170 PRINT "YOU WILL MARRY ";
180 LET C=INT(RND(1)*4)*10:GOTO 190+C
190 PRINT NAME1$:END
200 PRINT NAME2$:END
210 PRINT NAME3$:END
220 PRINT NAME4$:END
230 END
240 PRINT "NAME ";:INPUT TEXT$
250 PRINT
260 RETURN
```

אורים ותומים

הנה תכנית שתראה לכל החברים שלך איזה מכשיר חכם יש לך ! אפשר להכין מראש כל מיני תשובות (שורות 101-104). אם יש ברצונך להוסיף תשובות, אפשר להכניסן בסדר עולה החל משורה 105. שנוי בשורה 45 יביא לכך שהמספר האקראי שנבחר (CHOICE) יתאים למספר המדויק של תשובות הנמצאות

במשפטי DATA. למשל, אם רוצים סך הכל 10 תשובות, אזי משפטי ה DATA יהיו משורות 101-110, ושורה 45 תהיה כדלקמן :

```
45 CHOICE=INT(RND(1)*10)+1
10 REM *** ORACLE ***
15 REM HERB KOHL & LEN LINDSAY, 1980
20 DIM QUESTION$(80),ANSWER$(40)
25 PRINT "[CLEAR]"
30 PRINT "WHAT QUESTION DO YOU HAVE
   FOR":PRINT "THE ORACLE"
40 INPUT QUESTION$
45 CHOICE=INT(RND(1)*4)+1
50 RESTORE 100+CHOICE
60 READ ANSWER$
70 PRINT
75 PRINT "THE ORACLE SAYS,"
80 PRINT ANSWER$
85 PRINT :PRINT
90 GOTO 30
101 DATA "IT ALL DEPENDS ON YOU!"
102 DATA "ONLY IF YOU ASK..."
103 DATA "YES--BUT ONLY ONCE!"
104 DATA "IF YOU DON'T TELL"
```

ההורוסקופ האישי שלך- ממוחשב

זהו נוסח שונה במקצת של שתי התכניות הקודמות - מגיד העתידות והאורים ותומים. כל אחד יכול למלא את ההורוסקופ האישי שלו לכל סימן ע"י שנוי תוכן השורות במשפטי DATA בשורות 250-140. אולם, ההורוסקופים שלך יוכלו להיות בני שתי שורות בלבד (סה"כ 80 תוים).
הסיבה לכך היא שהלולאה משורות 90-120 בודקת באופן שטתי כל משפט DATA כדי לראות אם הוא מתאים לסימן שלך (או לפחות לשלש האותיות הראשונות שלו !)

```
10 REM *  <<< ASTROLOGY PROGRAM >>>
20 REM *      copyright (C) 1980
30 REM *   by Kohl, Kahn & Lindsay
40 DIM SIGN$(10),TEXT$(80)
50 PRINT "[CLEAR]"
```

```
60 PRINT :PRINT :PRINT :PRINT "WHAT IS Y
OUR SIGN";
70 INPUT SIGN$
80 RESTORE :PRINT
90 FOR LOOP=1 TO 12
100 READ TEXT$
110 IF TEXT$(1,3)=SIGN$(1,3) THEN PRINT
TEXT$:POP :GOTO 60
120 NEXT LOOP
130 PRINT "I DO NOT KNOW THAT SIGN":GOTO
 60
140 DATA LEO:  YOU THRIVE ON FUN AND GAM
ES
150 DATA ARIES: YOU AREN'T TOO BASHFUL A
BOUT    GOING AFTER WHAT YOU WANT
160 DATA LIBRA: DON'T OVERDO THE PLEASUR
E--LOOK  OUT FOR WORK ADVANCES TOO
170 DATA VIRGO: YOU'RE COOL AS A CUCUMBE
R BUT   YOU NEED SOMEONE TO CONFIDE IN
180 DATA CANCER:  DON'T LET YOUR EMOTION
AL UPS AND DOWNS GET TO YOU
190 DATA GEMINI: YOU'RE NEVER ALONE--YOU
R CHARM  WILL TAKE CARE OF YOU
200 DATA PISCES:  LOOK BEYOND THE SURFAC
E WORD  OR TOUCH
210 DATA TAURUS:  YOU'RE POSSESSIVE BUT
DON'T  LET IT KEEP YOU DOWN
220 DATA SCORPIO:  DON'T BE GUILTY--JUST
 BE     GRATEFUL (AND SHOW IT)
230 DATA AQUARIUS: YOU ALWAYS SURPRISE E
VERYONE SO MAYBE NOW IT'S YOUR TURN
240 DATA CAPRICORN:  TAKE RISKS THAT APP
EAL TO  YOUR SENSE OF GROWTH
250 DATA SAGITARIUS:  YOU'RE DYNAMITE SO
 GO TO IT NOW
```

עוגיות מזל!

עוגיות מזל הן עוגיות שבהן מכניסים פתק אישי עם הודעה על חיזוי העתיד. זהו מנהג סיני, הגורם התרגשות והנאה. לפנינו נוסח מעובד של תכנית ההורוסקופ. (כדאי שהמחשב ידפיס אותן באמצעות המדפסת !) אפשר לקחת את המרשמים מספר בשול, ולהגיש את העוגיות לחברים ולבני המשפחה. תכנית ההורוסקופ ניתנת לשינוי וגוון די בקלות, כך שתייצר בשבילך עוגידת מזל !

שאלות טיפשיות,תשובות טיפשיות

זוהי תכנית שמדגימה את הפתגם העתיק של המחשב :
אם נכנס זבל, יוצא זבל.
יחד עם זאת, אפשר לנצל זאת במסיבות, כשרוצים להראות לחברים עד כמה באמת חכם המחשב שלך !

```
10 REM * < SILLY QUESTIONS/ANSWERS >
20 REM *  (C) 1980 by Herbert Kohl,
30 REM *  Ted Kahn and Len Lindsay
40 REM *    Demonstrates use of
50 REM *        GOSUB & RETURN
60 DIM QUESTION1$(80),QUESTION2$(80),QUE
STION3$(80),A$(7)
70 DIM ANSWER1$(80),ANSWER2$(80),ANSWER3
$(80),TEXT$(80)
80 PRINT "[CLEAR]"
90 A$="A":GOSUB 210:QUESTION1$=TEXT$
100 A$="ANOTHER":GOSUB 210:QUESTION2$=TEXT$
110 GOSUB 210:QUESTION3$=TEXT$
120 PRINT "[CLEAR]"
130 A$="A":GOSUB 250:ANSWER1$=TEXT$
140 A$="ANOTHER":GOSUB 250:ANSWER2$=TEXT
$
150 GOSUB 250:ANSWER3$=TEXT$
160 PRINT "[CLEAR]"
170 PRINT QUESTION1$;"?":GOSUB 290
180 PRINT QUESTION2$;"?":GOSUB 290
190 PRINT QUESTION3$;"?":GOSUB 290
200 END
210 PRINT "TYPE IN ";A$;" SILLY QUESTION
.":REM MUST BE LESS THAN TWO LINES LONG
220 INPUT TEXT$
230 PRINT
240 RETURN
250 PRINT "TYPE IN ";A$;" SILLY ANSWER"
260 INPUT TEXT$
270 PRINT
280 RETURN
290 CHOICE=INT(RND(1)*3)+1
300 IF CHOICE=1 THEN PRINT ANSWER1$
310 IF CHOICE=2 THEN PRINT ANSWER2$
320 IF CHOICE=3 THEN PRINT ANSWER3$
330 PRINT
340 RETURN
```

צבע

זוהי בעצם תכנית נוספת של אורים ותומים, אלא שהפעם יש לה איזה פרס קטן : המחשב משנה את צבע המסך על פי ההעדפה האישית שלך. (הסבר הרבה יותר יסודי על השמוש בצבע נביא בפרק ח').

```
10 REM *  <<< COLOR PREFERENCE >>>
12 REM *     copyright (C) 1980 by
14 REM *     Kahn, Kohl, & Lindsay
16 DIM TEXT$(30),MESSAGE$(30)
18 PRINT "[CLEAR]"
20 SETCOLOR 1,1,0
22 SETCOLOR 2,0,10
24 PRINT "WHICH COLOR DO YOU LIKE MOST"
26 PRINT "RED, BLUE, OR YELLOW"
28 TEXT$="":NC=0
30 INPUT TEXT$
32 TRAP 40
34 IF TEXT$(1,1)="R" THEN NC=1:GOTO 42
36 IF TEXT$(1,1)="B" THEN NC=2:GOTO 42
38 IF TEXT$(1,1)="Y" THEN NC=3:GOTO 42
40 PRINT "[CLEAR]PLEASE CHOOSE ONLY ":GO
TO 26
42 RESTORE 100+ASC(TEXT$)
44 PRINT "[CLEAR]";
46 READ MESSAGE$
48 ON NC GOSUB 200,210,220
50 PRINT :PRINT "YOU ARE ";MESSAGE$
52 PRINT :PRINT "NEXT PERSON, PLEASE."
54 GOSUB 300
56 GOSUB 300:GOTO 18
166 DATA PLACID AND PEACEFUL
182 DATA HOTBLOODED AND INDEPENDENT
189 DATA IMPATIENT BUT OPTIMISTIC
200 SETCOLOR 2,3,4:RETURN
210 SETCOLOR 2,8,2:RETURN
220 SETCOLOR 2,1,12:RETURN
300 WAIT=600:REM DEFAULT WAIT LENGTH
310 FOR LOOP=1 TO WAIT
320 NEXT LOOP
330 RETURN
```

אלה כמה תכניות יותר מורכבות. הראשונה, שאנו קוראים
לה USSTATES, יכולה לעזור בלימוד הגיאוגרפיה ושמות
המדינות השונות של ארצות הברית. זהו נוסח דומה
לתכנית הסדרות שהבאנו קודם.

```
10 REM <<< USSTATES >>>
20 REM LEN LINDSAY, 1980
30 NUMBER=50
40 DIM PLAYER1$(8),PLAYER2$(8)
50 DIM IN$(15),LASTW$(14),STARTL$(1),TES
T$(14)
60 DIM PICKED(NUMBER),S(1),P(1),T(1)
70 GAMEOVER=9000
80 OPTION=2110:GOSUB OPTION-10
90 GOSUB 2000:REM GRAPHICS & COLORS
100 PRINT #6;"    STATE  NAMES"
110 POSITION 6,8:PRINT #6;"TO BEGIN"
120 POSITION 5,9:PRINT #6;"HIT  start"
130 POSITION 2,2:PRINT #6;"FOR INSTRUCTI
ONS"
140 POSITION 5,3:PRINT #6;"HIT option"
150 POSITION 6,11:PRINT #6;" ATARI "
160 KEY=PEEK(53279)
170 IF KEY=3 THEN GOSUB 1000:GOTO 90
180 IF KEY=5 THEN GOSUB OPTION:OPTION=OP
TION+10:GOSUB 2220:IF OPTION>2130 THEN O
PTION=2100
190 IF KEY=6 THEN 250
200 POSITION 0,5:PRINT #6;PLAYER1$;" VS
";PLAYER2$;"HIT select TO CHANGE"
210 SETCOLOR 3,0,RND(1)*256
220 FOR DELAY=1 TO 50:NEXT DELAY
230 SETCOLOR 2,0,PN:PN=PN+2
240 GOTO 160
250 GRAPHICS 2+16:PRINT #6;" WAIT"
260 FOR A=1 TO NUMBER:PICKED(A)=0:NEXT A
:REM INIT
270 S(0)=0:S(1)=0
280 T(0)=0:T(1)=0
290 COUNT=0
300 PLAYER=1
310 TEST$=PLAYER2$:GOSUB 2300:PLAYER2$=T
EST$
320 PASS=0
330 PLAYER=1-PLAYER
```

```
340 POSITION 0,10:GOSUB 6900
350 POSITION 0,11:GOSUB 6900
360 IF PASS>1 THEN GOTO GAMEOVER
370 COUNT=COUNT+1
380 GOSUB 6000
390 POSITION 1,3:PRINT #6;"LAST WORD WAS
:"
400 POSITION 1,4:GOSUB 6900:REM ERASE LI
NE
410 IF LEN(LASTW$)>1 THEN IF LASTW$(1,1)
<="Z" THEN TEST$=LASTW$:GOSUB 2300:POSIT
ION 1,4:PRINT #6;TEST$
420 POSITION 1,6:PRINT #6;"NEXT WORD MUS
T"
430 POSITION 3,7:PRINT #6;"START WITH: "
;
440 STARTL$=CHR$(INT(RND(1)*26)+65)
450 IF LEN(LASTW$)>1 THEN STARTL$=LASTW$
(LEN(LASTW$),LEN(LASTW$))
460 PRINT #6;CHR$(ASC(STARTL$)+160)
470 POSITION 1,8
480 IF P(PLAYER)=0 THEN PRINT #6;"ENTER
YOUR WORD"
490 IF P(PLAYER)=1 THEN PRINT #6;"MY WOR
D IS:     "
500 GOSUB 6900
510 IF P(PLAYER)=1 THEN GOSUB 7000:GOTO
600:REM COMPUTER TURN
520 CHAR=1
530 IN$="                    "
540 POSITION 1,9:PRINT #6;IN$
550 OPEN #1,4,0,"K"
560 POKE 764,255:REM CLEAR BUFFER
570 IF PEEK(764)=255 THEN SETCOLOR 3,0,R
ND(1)*256:GOTO 570
580 GET #1,IN
590 CLOSE #1
600 IF IN=126 AND CHAR>1 THEN CHAR=CHAR-
1:IN$(CHAR,CHAR)=" ":GOTO 540:REM DELETE HIT
610 IF IN=155 THEN 670:REM RETURN HIT
620 IF IN<65 AND IN<>32 OR IN>90 THEN 55
0:REM NOT ALPHABETIC
630 IN$(CHAR,CHAR)=CHR$(IN)
640 CHAR=CHAR+1
```

```
650 IF CHAR>15 THEN CHAR=15
660 GOTO 540
670 IF LEN(IN$)<1 THEN 520
680 IF CHAR>1 THEN IN$=IN$(1,CHAR-1):REM
 MINUS TRAILING SPACES
690 IF IN$="PASS" THEN PASS=PASS+1:GOTO
330
700 IF ASC(IN$)<>ASC(STARTL$) THEN PASS=
PASS+1:PRINT #6;"1st letter's not ";STAR
TL$:GOTO 840
710 RESTORE
720 FOR TEST=1 TO NUMBER
730 READ TEST$
740 IF TEST$=IN$ THEN POP :GOTO 780
750 NEXT TEST
760 PRINT #6;"not a legal word"
770 GOTO 840:REM CONTINUE
780 IF PICKED(TEST)>0 THEN PRINT #6;"tha
t word is used   hit start":GOSUB 2200:G
OSUB 6000:GOTO 340
790 PICKED(TEST)=PLAYER+1
800 PRINT #6;"that word works"
810 LASTW$=IN$
820 S(PLAYER)=S(PLAYER)+1
830 T(PLAYER)=T(PLAYER)+LEN(IN$)
840 PRINT #6;"HIT start";
850 GOSUB 6100
860 GOSUB 2200:IF KEY<>6 THEN 860
870 GOSUB 7100:GOTO 320
999 END
1000 GRAPHICS 1+16:GOSUB 2010
1010 PRINT #6;"NAMES OF STATES ARE"
1020 PRINT #6;"THE LEGAL WORDS."
1030 PRINT #6;"THE LAST LETTER OF"
1040 PRINT #6;"THE PREVIOUS WORD"
1050 PRINT #6;"MUST BE THE FIRST"
1060 PRINT #6;"LETTER OF THE NEXT"
1070 PRINT #6;"WORD. NO REPEATS"
1075 PRINT #6
1080 PRINT #6;"type 'PASS' if you"
1090 PRINT #6;"can not move."
1095 PRINT #6
1100 PRINT #6;"AN ILLEGAL MOVE IS"
1110 PRINT #6;"TREATED AS PASSING"
```

```
1120 PRINT #6;"YOUR TURN"
1125 PRINT #6
1130 PRINT #6;"THE GAME IS OVER"
1140 PRINT #6;"IF BOTH PLAYERS PASS"
1200 PRINT #6
1900 PRINT #6;"HIT start"
1990 GOSUB 2200:IF KEY<>6 THEN 1990
1999 RETURN
2000 GRAPHICS 2+16
2010 SETCOLOR 0,2,8:REM CAPS - REG TEXT
2020 SETCOLOR 1,12,10:REM lower LUMIN -
OPTIONS IN STARTER
2030 SETCOLOR 2,8,12:REM CAPS WINDOW - T
ITLE IN STARTER
2040 SETCOLOR 3,4,6:REM lower - NAMES OF
 THE SPECIAL KEYS - VIBRATES NAMES
2050 SETCOLOR 4,0,0:REM BACKGROUND
2099 RETURN
2100 PLAYER1$="human   ":PLAYER2$="compu
ter":P(0)=0:P(1)=1
2109 RETURN
2110 PLAYER1$="computer":P(0)=1
2119 RETURN
2120 PLAYER2$="human   ":P(1)=0
2129 RETURN
2130 PLAYER1$="human   ":P(0)=0
2139 RETURN
2200 IF PEEK(53279)<>7 THEN 2200
2210 KEY=PEEK(53279):IF KEY=7 THEN SETCO
LOR 3,0,RND(1)*255:GOTO 2210
2220 IF PEEK(53279)<>7 THEN 2220
2299 GOSUB 2010:RETURN
2300 FOR A=1 TO LEN(TEST$)
2310 TEST$(A,A)=CHR$(ASC(TEST$(A,A))+128)
2320 NEXT A
2399 RETURN
6000 GOSUB 2010
6010 TEST$=PLAYER1$:GOSUB 2300:PLAYER1$=
TEST$
6020 TEST$=PLAYER2$:GOSUB 2300:PLAYER2$=
TEST$
6030 POSITION 0,0
6050 PRINT #6;PLAYER1$;" VS ";PLAYER2$
6100 POSITION 1,1:PRINT #6;S(0);":";T(0)
```

```
6110 POSITION 13,1:PRINT #6;S(1);":";T(1
)
6900 PRINT #6;"                              ";
6999 RETURN
7000 RESTORE
7010 FOR TEST=1 TO NUMBER
7020 SETCOLOR 3,3,RND(1)*256
7030 READ IN$
7040 IF ASC(IN$)=ASC(STARTL$) AND PICKED
(TEST)=0 THEN POP :GOTO 7100
7050 NEXT TEST
7060 IN$="PASS"
7070 POSITION 1,10
7080 PRINT #6;"I PASS! HIT start"
7090 GOSUB 2200:IF KEY<>6 THEN 7090
7100 GOSUB 2300
7110 CHAR=LEN(IN$)+1
7120 POSITION 1,9:PRINT #6;IN$
7199 RETURN
9000 POSITION 1,10
9010 PRINT #6;"[CLEAR]"
9020 FOR L=30 TO 10 STEP -0.5:SOUND 0,L,
10,10:NEXT L:SOUND 0,0,0,0
9030 GOSUB 6000
9040 PRINT #6
9050 POSITION 3,10:PRINT #6;"GAME IS OVE
R"
9060 POSITION 4,11:PRINT #6;"HIT start";
9070 GOSUB 200:IF KEY<>6 THEN 9070
9080 RUN
10650 DATA ALABAMA,ALASKA,ARIZONA,ARKANS
AS
10660 REM B
10670 DATA CALIFORNIA,COLORADO,CONNECTICUT
10680 DATA DELAWARE
10690 REM E
10700 DATA FLORIDA
10710 DATA GEORGIA
10720 DATA HAWAII
10730 DATA IDAHO,ILLINOIS,INDIANA,IOWA
10740 REM J
10750 DATA KANSAS,KENTUCKY
10760 DATA LOUISIANA
10770 DATA MAINE,MARYLAND,MASSACHUSETTS,
```

```
MICHIGAN,MINNESOTA,MISSISSIPPI,MISSOURI,
MONTANA
10780 DATA NEBRASKA,NEVADA,NEW HAMPSHIRE
,NEW JERSEY,NEW MEXICO,NEW YORK,NORTH CA
ROLINA,NORTH DAKOTA
10790 DATA OHIO,OKLAHOMA,OREGON
10800 DATA PENNSYLVANIA
10810 REM Q
10820 DATA RHODE ISLAND
10830 DATA SOUTH CAROLINA,SOUTH DAKOTA
10840 DATA TENNESSEE,TEXAS
10850 DATA UTAH
10860 DATA VERMONT,VIRGINIA
10870 DATA WASHINGTON,WEST VIRGINIA,WISC
ONSIN,WYOMING
10880 REM X
10890 REM Y
10900 REM Z
10910 DATA ZZZ
```

וכעת, משחק בשם "טוסיץ" (TOESITCH). זהו משחק לשנים.
לכל שחקן יש רשימת מלים שממנה יש לבחור.
מטרת המשחק היא לבחור שלש מלים המכילות אותה האות.
הוראות מפורטות יותר מופיעות בתכנית עצמה.

```
10 REM <<< TOESITCH >>>
20 REM LEN LINDSAY 1980
30 DIM PLAYER1$(8),PLAYER2$(8),A$(1),P$(
4),W$(4),WORD$(4)
40 DIM P(2),W(9),L(1,26)
50 GOSUB 2000:REM SETCOLORS
60 GM=2110
70 GOSUB 2100:REM INIT PLAYERS
80 GAMEOVER=9000
90 WIN=99:REM INIT
100 PRINT #6;"  toes itch game"
110 GOSUB 4400:REM INIT
120 POSITION 6,8:PRINT #6;"TO BEGIN"
130 PRINT #6;"      HIT  START"
140 POSITION 2,2:PRINT #6;"FOR INSTRUCTI
ONS"
150 PRINT #6;"      HIT OPTION"
160 POSITION 6,11:PRINT #6;" ATARI"
```

```
170 KEY=PEEK(53279)
180 IF KEY=3 THEN GOSUB 1000:GOTO 100
190 IF KEY=5 THEN GOSUB GM:GM=GM+10:GOSU
B 2520:IF GM>2130 THEN GM=2100
200 IF KEY=6 THEN 290
210 POSITION 0,5:PRINT #6;PLAYER1$;" VS
";PLAYER2$;"HIT SELECT TO CHANGE"
220 SETCOLOR 2,0,RND(1)*256
230 FOR DELAY=1 TO 150:NEXT DELAY
240 GOTO 170
250 REM
260 REM -TO CHANGE WORDS, JUST CHANGE TH
E WORDS IN DATA IN LINE 280
270 REM
280 DATA SLIM,WORM,GAME,ITCH,TOES,TALK,V
INE,PONY,SAND
290 FOR A=1 TO 9:W(A)=1:NEXT A
300 PLAYER=2
310 WORDLEFT=9
320 GOSUB 2000
330 GOSUB 6000
340 PLAYER=2-PLAYER:REM SWITCH PLAYERS
350 IF WORDLEFT=0 THEN GOTO GAMEOVER
360 GOSUB 2010
370 IF P(PLAYER)=1 THEN GOSUB 3000:REM H
UMAN
380 IF P(PLAYER)=0 THEN GOSUB 7000:REM C
OMPUTER
390 GOSUB 4500:REM SEE IF 3 LETTERS SAME
400 GOTO 340
999 END
1000 GRAPHICS 1+16
1030 PRINT #6;"TWO PLAYERS TAKE"
1040 PRINT #6;"TURNS PICKING A"
1050 PRINT #6;"WORD FROM A LIST"
1060 PRINT #6;"IN THE CENTER OF"
1070 PRINT #6;"THE SCREEN."
1075 PRINT #6
1080 PRINT #6;"select=change word"
1090 PRINT #6;"start=take that word"
1100 PRINT #6;"THE LETTERS IN EACH"
1110 PRINT #6;"WORD ARE THE MAJOR"
1120 PRINT #6;"FOCUS OF THIS GAME."
1125 PRINT #6
```

```
1130 PRINT #6;"the first player to"
1140 PRINT #6;"have three of any"
1150 PRINT #6;"letter wins."
1155 PRINT #6
1160 PRINT #6;"THE GAME IS A DRAW"
1170 PRINT #6;"IF ALL WORDS ARE"
1180 PRINT #6;"PICKED AND NO ONE"
1190 PRINT #6;"WON YET."
1195 PRINT #6
1200 PRINT #6;"HIT start TO RETURN"
1900 GOSUB 2500:IF KEY<>6 THEN 1900
1999 RUN
2000 GRAPHICS 2+16
2010 REM
2011 SETCOLOR 3,4,4:REM RED
2012 SETCOLOR 2,0,14:REM WHITE
2013 SETCOLOR 1,8,6:REM BLUE
2014 SETCOLOR 4,0,0:REM BLACK
2015 SETCOLOR 0,2,8:REM GOLD
2099 RETURN
2100 PLAYER1$="computer":PLAYER2$="human
   ":P(0)=0:P(2)=1:RETURN
2110 PLAYER2$="computer":P(2)=0:RETURN
2120 PLAYER1$="human    ":P(0)=1:RETURN
2130 PLAYER2$="human    ":P(2)=1:RETURN
2500 IF PEEK(53279)<>7 THEN 2500
2510 KEY=PEEK(53279):IF KEY=7 THEN SETCO
LOR PLAYER+1,0,RND(1)*255:FOR WW=1 TO 50
:NEXT WW:GOTO 2510
2520 IF PEEK(53279)<>7 THEN 2520
2599 RETURN
3000 WORDUP=INT(RND(1)*9+1):REM INIT
3010 IF WORDUP>9 THEN WORDUP=1
3020 IF W(WORDUP)<>1 THEN WORDUP=WORDUP+
1:GOTO 3010
3030 GOSUB 6000
3040 IF P(PLAYER)=0 THEN 3080
3050 GOSUB 2510
3060 IF KEY=5 THEN POSITION 7,WORDUP:WOR
DUP=WORDUP+1:GOTO 3010
3070 IF KEY<>6 THEN 3030
3080 W(WORDUP)=PLAYER:GOSUB 6000
3090 WORDLEFT=WORDLEFT-1
3099 RETURN
```

```
4000 REM P$ IN AND P$ OUT (+128)
4010 FOR DEL=1 TO 4
4020 P$(DEL,DEL)=CHR$(ASC(P$(DEL,DEL))+1
28)
4030 NEXT DEL
4099 RETURN
4100 REM HIGHLIGHT THE WINNING LETTER
4101 REM WORDS ARE W$
4102 REM WINNING LETTER IS A$
4110 FOR KEY=1 TO 4
4120 IF W$(KEY,KEY)=A$ THEN W$(KEY,KEY)=
CHR$(ASC(A$)+128)
4130 NEXT KEY
4199 RETURN
4400 FOR DEL=0 TO 1
4410 FOR KEY=1 TO 26
4420 L(DEL,KEY)=0
4430 NEXT KEY
4440 NEXT DEL
4499 RETURN
4500 REM UPDATE LETTER COUNT
4501 REM WORD IS W(WORDUP)
4510 FOR KEY=1 TO 4
4520 LTR=ASC(WORD$(KEY,KEY))-64
4530 L(PLAYER*0.5,LTR)=L(PLAYER*0.5,LTR)
+1
4540 IF L(PLAYER*0.5,LTR)=3 THEN POP :PO
P :A$=CHR$(LTR+64):WIN=PLAYER:GOTO GAMEO
VER
4550 NEXT KEY
4599 RETURN
6000 POSITION 0,0
6010 PRINT #6;PLAYER1$
6020 POSITION 12,0
6030 PRINT #6;PLAYER2$
6100 RESTORE
6110 FOR ROW=1 TO 9
6120 READ W$
6130 FOR VERT=0 TO 2
6140 POSITION VERT*7+1,ROW
6150 P$="      "
6160 IF W(ROW)=VERT THEN P$=W$:IF ROW=WO
RDUP THEN WORD$=W$:GOSUB 4000
6170 PRINT #6;P$;
```

```
6180 NEXT VERT
6190 NEXT ROW
6199 RETURN
7000 IF W(5)=1 THEN WORDUP=5:GOSUB 3010:
GOTO 7999
7010 GOSUB 3000
7999 RETURN
9000 REM GAMEOVER
9010 RESTORE
9020 FOR DEL=1 TO 9
9030 READ W$
9040 GOSUB 4100
9050 IF W(DEL)=PLAYER THEN POSITION PLAY
ER*7+1,DEL:PRINT #6;W$
9060 NEXT DEL
9070 FOR LL=30 TO 1 STEP -0.2:SOUND 0,LL
,10,10:NEXT LL:SOUND 0,0,0,0
9100 POSITION 4,10
9110 IF WIN<3 THEN PRINT #6;"THREE ";CHR
$(LTR+192);" WINS"
9120 IF WIN>2 THEN PRINT #6;"NO ONE WINS
":WORDUP=0:GOSUB 6000
9200 POSITION 6,11:PRINT #6;"HIT START";
9210 ZW=1
9220 SETCOLOR 2,RND(1)*16,4
9230 IF WIN>2 THEN 9260
9240 ZW=1-ZW:SETCOLOR PLAYER+1,1,ZW*RND(
1)*255
9250 FOR DEL=1 TO 50:NEXT DEL
9260 IF PEEK(53279)=7 THEN 9220
9999 RUN
```

ולבסוף, תכנית של משחק אוצר מלים שבו יש לתקתק את המלים שרוצים ללמוד כ DATA (החל משורה 1000).
נוסף למילה שאותה צריך לנחש, מתקתקים "רמזים" לשחקן; רמזים אלה הם : מלה נרדפת, הגדרת המלה ; איזו הברה מדגישים; וכמה תנועות יש במלה.
כל משפט DATA משורה 1000 ואילך מכיל את המלה וכן את כל אחד מן הרמזים הללו, מופרדים ע"י פסיק.
(התכנית כבר מכילה ארבע דוגמאות, בשורות 1000, 1010, 1020 ו-1030). מאחר שהתכנית הזו מאפשרת לך להכניס את המלים שלך, הרי שאפשר להשתמש בה ללמוד מלים באנגלית.

```
10 REM * <<< WANTED: VOCABULARY >>>
20 REM *    (C) 1980 Len Lindsay
30 REM *          idea from
40 REM *    Calculators/Computers
50 PRINT "ADD YOUR OWN WORDS IN LINES 10
00 ON ":FOR WAIT=1 TO 1000:NEXT WAIT
60 DIM PART$(18),WORD$(18),SYNONYM$(18),
MEANING$(36),ACCENT$(18),SYLABLE$(10),VO
WEL$(10),CONSONANT$(10),TEXT$(18)
70 DIM XX$(18)
80 DIM NAME$(18)
90 POKE 82,2:REM SET LEFT MARGIN
100 SETCOLOR 2,2,6
110 SETCOLOR 4,2,6
120 SETCOLOR 1,2,2
130 REM
140 WORDS=0:REM INITIALIZE - THEN COUNT
HOW MANY WORDS ARE AVAILABLE
150 RESTORE 1000+10*(WORDS+1)
160 TRAP 200:REM THIS LETS US CONTINUE W
HEN WE RUN OUT OF DATA
170 READ PART$:REM IF NOTHING LEFT TO RE
AD -ERROR- AND GOTO TRAP 200
180 WORDS=WORDS+1:REM INCREASE COUNT
190 GOTO 150:REM SEE IF MORE WORDS
200 PICK=INT(RND(1)*WORDS)+1:REM CHOOSE
A RANDOM WORD TO USE
210 RESTORE 1000+PICK*10:REM RESTORE THE
 DATA TO START AT CHOSEN WORD
220 READ PART$
230 READ WORD$
240 READ SYNONYM$
250 READ MEANING$
260 READ SYLABLE
270 READ ACCENT
280 READ VOWEL
290 RESTORE 900+SYLABLE:READ SYLABLE$
300 RESTORE 900+ACCENT:READ TEXT$:READ A
CCENT$
310 RESTORE 900+VOWEL:READ VOWEL$
320 RESTORE 900+LEN(WORD$)-VOWEL:READ CO
NSONANT$
330 GOSUB 6000
340 POSITION 20,1:PRINT PART$
```

```
350 POSITION 20,3:PRINT SYNONYM$
360 CLUES=0:REM INITIALIZE CLUE COUNT
370 POSITION 2,18:REM READY FOR QUESTION
S
380 PRINT "[DEL LINE][DEL LINE][DEL LINE
][DEL LINE][DEL LINE][DEL LINE][DEL LINE
][DEL LINE]";:REM DELETE LINES FROM BOTT
OM OF SCREEN
390 PRINT "WHAT DO YOU THINK THE WORD IS
?"
400 GOSUB 15000:REM INPUT ROUTINE
410 IF XX$=WORD$ THEN 750
420 PRINT "YOU DID NOT IDENTIFY THE WORD
!";
430 PAUSE=100:GOSUB 14020:REM PAUSE
440 CLUE=CLUE+1
450 ON CLUE GOSUB 470,530,600,680
460 GOTO 370
470 POSITION 21,5
480 PRINT "[.][LEFT]";
490 GOSUB 14000:REM PAUSE
500 PRINT "[BELL]";
510 PRINT SYLABLE$
520 RETURN
530 POSITION 21,11
540 PRINT "[.][LEFT]";
550 GOSUB 14000:REM PAUSE
560 PRINT "[BELL]";:REM BEEP
570 IF ACCENT$="NONE" THEN ACCENT$="ONLY
 ONE SYLABLE"
580 PRINT ACCENT$
590 RETURN
600 POSITION 21,7
610 PRINT "[.][LEFT]";
620 GOSUB 14000:REM PAUSE
630 PRINT "[BELL]";:REM BEEP
640 PRINT VOWEL$
650 POSITION 21,9
660 PRINT CONSONANT$
670 RETURN
680 CLUE=3
690 PRINT "NO MORE CLUES, BUT WATCH CLOS
ELY AND"
700 PRINT "YOU WILL GET A PEEK OF THE WORD"
```

```
710 PRINT "             ";WORD$;"[UP]"
720 GOSUB 14000:REM PAUSE
730 PRINT "[DEL LINE]";
740 RETURN
750 PRINT "YES! THAT IS CORRECT!"
760 CASE=CASE+1
770 PRINT "YOU HAVE NOW SOLVED ";CASE;"
CASE";:IF CASE>1 THEN PRINT "S";
780 PRINT
790 PRINT "WOULD YOU LIKE ANOTHER CASE?"
;
800 GOSUB 15100:REM GET ROUTINE
810 IF XX$="N" THEN END
820 IF XX$="Y" THEN 200
830 GOTO 800
840 REM DATA TO CHANGE 1 TO ONE OR FIRST
 ETC...
900 DATA NONE,NONE
901 DATA ONE,FIRST
902 DATA TWO,SECOND
903 DATA THREE,THIRD
904 DATA FOUR,FOURTH
905 DATA FIVE,FIFTH
906 DATA SIX,SIXTH
907 DATA SEVEN,SEVENTH
908 DATA EIGHT,EIGHTH
909 DATA NINE,NINTH
910 DATA TEN,TENTH
999 END
1000 REM PART / WORD / SYNONYM / MEANING
 / SYLABLES / ACCENTED SYLABLE / VOWELS
1001 REM IF ONLY ONE SYLABLE - SET ACCEN
TED SYLABLE TO 0
1010 DATA ADJECTIVE,CENTRAL,MIDDLE,NEAR
THE CENTER,2,1,2
1020 DATA VERB,FORD,PASS THROUGH,CROSS W
ATER BY WADING,1,0,1
1030 DATA NOUN,SCYTHE,BLADE,LONG CURVED
BLADE ON LONG HANDLE,1,0,2
6000 PRINT "[CLEAR]";:REM CLEAR SCREEN
6100 PRINT "[Q][R][R][R][R][R][R][R][R][
R][R][R][R][R][R][R][R][R][R][R][R][R][R
][R][R][R][R][R][R][R][R][R][R][R][R][R]
[E]"
```

```
6110 PRINT "|            WANTED:
        |"
6120 PRINT "|
        |"
6130 PRINT "|              ALIAS:
        |"
6140 PRINT "|
        |"
6150 PRINT "|          SYLABLES:
        |"
6160 PRINT "|
        |"
6170 PRINT "|             VOWELS:
        |"
6180 PRINT "|
        |"
6190 PRINT "|        CONSONANTS:
        |"
6200 PRINT "|
        |"
6210 PRINT "|ACCENTED SYLABLE:
        |"
6220 PRINT "[A]------------------------
----------[D]"
6230 PRINT "|             LAST KNOWN MEANIN
G       |"
6240 PRINT "|
        |"
6245 PRINT "|
        |"
6250 PRINT "[UP]|";MEANING$
6255 PRINT "|
        |"
6260 PRINT "[Z][R][R][R][R][R][R][R][R][
R][R][R][R][R][R][R][R][R][R][R][R][R][R
][R][R][R][R][R][R][R][R][R][R][R][R][R]
[C]"
6999 RETURN
14000 REM *  PAUSE SUBROUTINE
14001 REM PAUSE IS DEFAULT AT 999
14002 REM TO SET YOUR OWN PAUSE
14003 REM SIMPLY DO THIS ON THE CALL
14004 REM
14005 REM PAUSE=300 : GOSUB 14020
```

```
14006 REM
14007 REM NOTE GOSUB AT 14020 IF YOU
14008 REM SET YOUR OWN PAUSE LENGTH
14009 REM
14010 PAUSE=999
14020 FOR LOOP=1 TO PAUSE
14030 NEXT LOOP
14099 RETURN
15000 REM *  INPUT SUBROUTINE
15001 REM XX$ IS STRING TO BE INPUT
15002 REM XX IS THE NUMERIC VALUE OF THE
 INPUT
15003 REM MUST DIM XX$(N)  -  N IS THE L
IMIT YOU WISH ON THE INPUT STRING LENGTH
15004 REM
15010 PRINT "[BELL]";
15020 POKE 764,255:REM CLEAR BUFFER
15030 TRAP 15030:REM TRAP ERROR ON INPUT
15040 INPUT XX$:REM INPUT STRING
15050 TRAP 15090:REM TRAP IF NO NUMERIC
AVAILABLE IN STRING
15060 XX=VAL(XX$):REM NUMERIC
15090 TRAP 34567:REM SPRING TRAP
15099 RETURN
15100 REM *  GET SUBROUTINE
15101 REM XX$ IS THE CHARACTER INPUT
15102 REM XX IS THE NUMERIC EQUIVALENT O
F INPUT CHARACTER
15103 REM USES FILE #4
15104 REM
15110 PRINT "[BELL]";
15120 POKE 764,255:REM CLEAR BUFFER
15130 TRAP 15130:REM TRAP ERROR ON GET
15139 CLOSE #4:REM MAKE SURE FILE IS CLO
SED BEFORE OPENING IT
15140 OPEN #4,4,0,"K:":REM OPEN KEYBOARD
 FOR A GET
15150 GET #4,XX:REM GET ASCII VALUE OF K
EY HIT
15155 CLOSE #4:REM CLOSE KEYBOARD FILE
15160 XX$=CHR$(XX):REM CONVERT TO STRING
 CHARACTER OF KEY HIT
15165 REM PRINT XX$;
15170 TRAP 15190:REM SET TRAP BEFORE NUM
```

```
ERIC CONVERSION
15175 XX=0:REM INITIALIZE TO ZERO INCASE
 NOT NUMERIC
15180 XX=VAL(XX$):REM SET XX AS THE DIGI
T HIT - 0 IF NONE
15190 TRAP 34567:REM SPRING TRAP
15199 RETURN
```



כמה תת-תכניות שימושיות למשחקי מילים

אנו מביאים כאן שתי תכניות שמדגימות פעולות יסודיות על מחרוזות אותיות, פעולות המהוות בסיס לרוב משחקי המלים. התכנית הראשונה מראה כיצד לבחור באופן אקראי אות מתוך האלפבית ולהדפיס אותה. (A$ הוא משתנה מחרוזת שהוגדר כאותיות האלפבית, לפי הסדר ; בוחרים מספר בין 1 ו-26 (שורה 30), והאות המיוצגת ע"י מספר זה -- A$(Y,Y), תודפס על המסך.

התכנית השניה מראה כיצד לבחור קבוצת אותיות מתוך מחרוזת. שורה 30 בוחרת רק אות אחת, כשהיא מתחילה (ומסימת) בתו השלישי במחרוזת (במקרה זה, האות C); שורה 40 מדפיסה את המחרוזת, מהתו הראשון עד התו העשירי.

```
5 REM GETTING RANDOM LETTER FROM LIST
10 DIM A$(26)
20 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
30 LET Y=INT(RND(1)*26)+1
40 PRINT A$(Y,Y)
```

```
5 REM HOW TO GET A LETTER OR
6 REM A GROUP OF LETTERS OUT OF A$
10 DIM A$(10)
20 A$="ABCDEFGHIJ"
30 PRINT A$(3,3)
40 PRINT A$(1,10)
```

אותיות גדולות

זוהי תת-תכנית שמאפשרת לך להציג על המסך אותיות גדולות. היא מתאימה במיוחד לילדים צעירים.

```
10 REM *       <<< BIGLETS >>>
20 REM *  (C) 1980 by Len Lindsay
30 REM * This routine shows how you
40 REM * can get your ATARI to display
50 REM * big letters on the screen
60 REM * as you type them.  Especially
70 REM * good for very young children
80 TRAP 160
90 REM *  For different size of
100 REM * letters use GRAPHICS 1+16
110 GRAPHICS 2+16
120 OPEN #1,4,0,"K"
130 GET #1,KEY
140 PRINT #6;CHR$(KEY);
150 GOTO 130
160 TRAP 160:GOTO 130
```

חלק שלישי

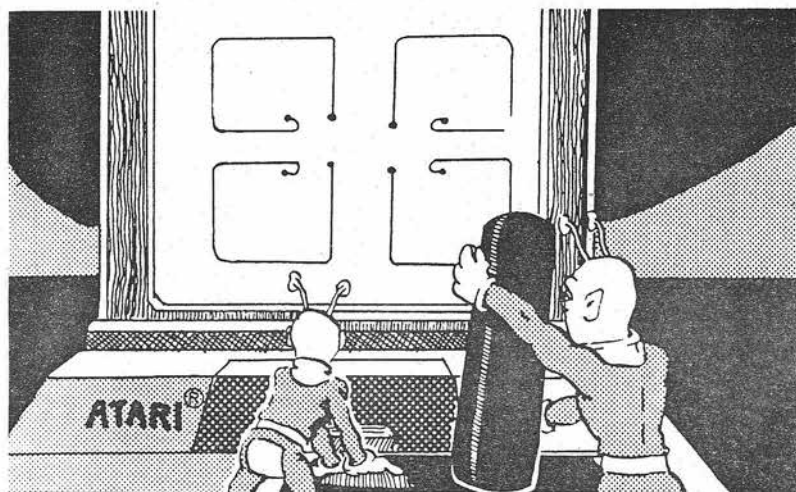
השרות המיוחד
של אטארי



למחשב האישי אטארי 400 או אטארי 800 יש כמה
תכונות מיוחדות שמאפשרות לך לצייר וליצור דמויות
נעות על המסך, לפקח על 128 צירופים שונים של צבע
ובהירות ולחבר מוסיקה תוך שמוש בתחום של 256 צלילים
לארבעה קולות. צבע,צליל וגרפיקה הם יתרונות מיוחדים
של אטארי. בחלק זה נראה לך כמה מהאפשרויות לשמוש
ולצירוף שלשת המימדים הללו ונביא כמה משחקים
הבנויים על צבע, קול ותנועה. הכרת התכניות שלנו
תתן לך את היכולת לגוון את המשחקים ולפתח חדשים.
מידע נוסף על צליל, צבע וגרפיקה נמצא במדריך
למשתמש של אטארי :

ATARI 400/800 BASIC Reference Manual

פרק ו'



גרפיקה

ה BASIC של אטארי מאפשר לך להשתמש בשלשה מצבי פעולה שונים עבור טקסט (מצבים 0-2) ובששה מצבים גרפיים שונים (מצבים 3-8) בכל מחשב אטארי 400 או 800. כל מצב גרפי הופך למעשה את מסך הטלביזיה שלך למעין נייר שרטוט שעליו ניתן לצייר.
ודאי שמת לב שיש גם **חלון** טקסט קטן בתחתית המסך, אשר בו אפשר להשתמש למתן הנחיות או להכנסת משתנים.
הנה סכום של מצבי הטקסט והמצבים הגרפיים של מחשב אטארי 800/400 שלך.

תכנית המסך

מצב גרפי	מה הוא מדפיס	אופקי (עמודות)	אנכי (שורות) מסך מפוצל	אנכי (שורות) מסך מלא
0	TEXT	40	-	24
1	TEXT	20	20	24
2	TEXT	20	10	12
3	GRAPHICS	40	20	24
4	GRAPHICS	80	40	48
5	GRAPHICS	80	40	48
6	GRAPHICS	160	80	96
7	GRAPHICS	160	80	96
8	GRAPHICS	320	160	192

בחלק הראשון הצגנו פקודות גרפיות כמו PLOT ו- DRAWTO, והדגמנו את השמוש בהן תוך כדי בנית המרוץ בין הארנבת והצב. אם יש ברצונך, אולי כדאי לחזור על הקטע ההוא לפני המשחקים וההדגמות שבפרק זה, כדי **לרענן** את הזכרון.

כדי להשתמש ב GRAPHICS 8, צריך המחשב שלך להיות בעל 16K RAM זכרון.

כמה משחקי שרטוט פשוטים

אלה כמה משחקים שיכולים לעזור לך להכיר את השיטה לשרטוט נקודות במצבים גרפיים שונים.
המשחק הראשון פשוט נותן לך לציין נקודה, ואז המחשב מצייר אותה עבורך. התכנית הזו משתמשת ב-GRAPHICS3, אבל עליך לכתוב אותה מחדש ולנסות אותה אותה גם עם מצבים גרפיים אחרים.

```
5 REM PICK A POINT AND PLOT IT
10 GRAPHICS 3:COLOR 1
20 PRINT "PICK A POINT AND PLOT IT"
25 PRINT "X=";:INPUT X
30 PRINT "Y=";:INPUT Y
40 PLOT X,Y
50 GOTO 20
```

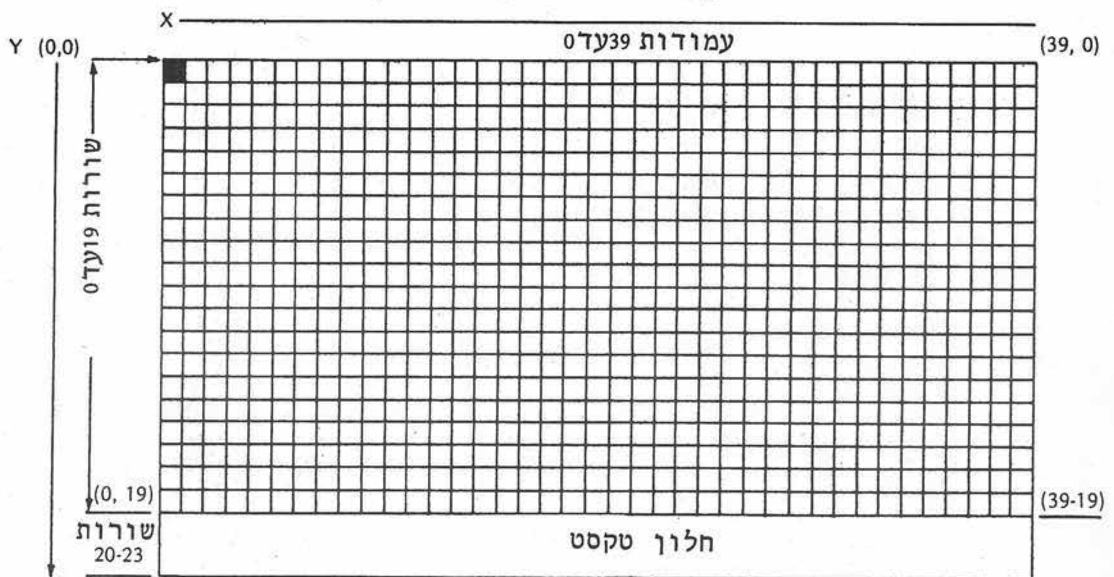
בגלל הפקודה GOTO בשורה 50, אפשר להמשיך ולשרטט נקודה אחר נקודה. אפשר לנסות לצייר תמונות באמצעות התכנית הפשוטה הזו, ולהתחיל בפנים או בדמות פשוטה.

והנה משחק שהוא ההפך של התכנית הקודמת המחשב בוחר נקודה, מצייר אותה, ומבקש ממך לנחש את נקודות הציון (הקואורדינטות) שלה. התכנית פשוטה, אבל דרוש די הרבה נסיון כדי לנחש בהצלחה את הקואורדינטות של נקודה מסוימת.

```
10 REM SIMPLE GRAPHICS LEARNING GAME
20 GRAPHICS 3:COLOR 1
30 LET X=INT(RND(1)*39)
40 LET Y=INT(RND(1)*19)
50 PRINT "HERE IS A POINT. GUESS ITS COORDINATES."
60 PLOT X,Y
70 PRINT "X=";:INPUT W
75 PRINT "Y=";:INPUT Z
80 IF X=W AND Y=Z THEN PRINT "CORRECT":GOTO 100
90 PRINT "TAKE ANOTHER GUESS":GOTO 50
100 FOR WAIT=1 TO 200:NEXT WAIT:GOTO 20
```

צריך לזכור שבמצב GRAPHICS3 צורת ארגון המסך היא כזו :





והנה עוד תכנית שתעזור לך להכיר את הרשת הגרפית. המחשב נותן לך את נקודות ה X וה Y, ועליך לנחש היכן הנקודה. אחרי השהיה קצרה מצייר המחשב את הנקודה על המסך, כך שיש ביכלתך לתקן את עצמך.

```
5 REM GUESS WHERE THE POINT IS
10 GRAPHICS 3:COLOR 1
20 LET X=INT(RND(1)*39)
30 LET Y=INT(RND(1)*19)
40 PRINT "GUESS WHERE THIS POINT IS"
50 PRINT X;",";Y
60 PRINT "POINT TO WHERE YOU THINK IT IS"
70 FOR M=1 TO 3500:NEXT M
```



```
80 PLOT X,Y
90 FOR N=1 TO 2500:NEXT N
100 PRINT "[CLEAR]"
110 GOTO 20
```

וזו תכנית יותר מתוחכמת לאותו המשחק. בגרסה זו, כאשר נראה שיודעים היכן תהיה הנקודה על המסך, עליך ללחוץ על כל מקש שהוא על לוח המקשים, וכן על מקש כלשהו כדי לקבל את הנקודה הבאה. אם רוצים מסך נקי לכל נקודה, צריך לשנות את שורה 80 ל :

```
80 GOTO 10
```

```
1 REM A SOPHISTICATED WAY TO . PROGRAM PLOT-
  TING GUESSING GAMES
2 REM OPEN-GET LEARNING, TOO
5 OPEN =1,4,0, "K:"
10 GRAPHICS 3
15 COLOR 1
20 X=INT(RND(1)*39)
25 Y=INT(RND(1)*19)
27 ?X;",";Y
30 ? "GUESS WHERE IT WILL BE"
40 GET =1,Q
50 PLOT X,Y
60 GET =1,Z
70 ? "[CLEAR]"
80 GOTO 20
```

מרבן, או נחשו את המרכז

אחת הדרכים הטובות ביותר ללמוד יותר על המחשב שלך היא לכתוב משחקים שיעזרו לך ללמוד מושגים חדשים בתכנות.

המשחק הזה, שנכתב ע"י לן לינדסי, הוא דרך נהדרת לתרגל את מערכת הקואורדינטות שבשמוש המצבים הגרפיים השונים באטארי. התכנית משתמשת רק ב GRAPHICS3, אבל אפשר לשנות אותה ולהתאימה לתרגול המצבים הגרפיים האחרים.

? הוא צורה נוספת של פקודת PRINT ב BASIC של אטארי. אנו משתמשים בה כי היא קצרה יותר, וחוסכת תקתוק.

במשחק הזה עליך לנחש את מקום ה X (העמודה) ומקום ה Y (השורה) של מרכז המלבן המוצג על המסך.
יש לתכנית שני מצבי פעולה. בראשון, ניתנים רמזים, או "היזון חוזר" אחרי כל ניחוש כך שאפשר לצמצם את טווח הניחוש. בסוף ניחוש מוצלח אפשר לכבות את הרמז או ה"ציונים" של ההיזון החוזר, ע"י לחיצה על המקש OPTION. (התכנית גם מרשה לנחש שוב).

התכנית עוזרת לילדים (ולמבוגרים) ללמוד לאמוד תשובה נכונה, על בסיס של מספר מינימלי של נתונים. זוהי מיומנות חיובית להצלחה בפתרון בעיות, במיוחד כאשר סדרי גודל ("עד כמה התשובה קרובה למציאות")הם יותר חשובים מן התשובה המדויקת.

הצעה לגוון נוסף של המשחק היא לתת לשחקן הערה שונה אם הצליח לנחש במספר נחושים קטן. כך יכולות להיות לך רמות שונות להשגים שונים, כמו למשל :

2 - 1 נחושים	מעריך מצויין
4 - 3 נחושים	מעריך
6 - 5 נחושים	מתלמד

וכו'

```
10 REM *  <<< GUESS THE CENTER >>>
20 REM *    (C) 1980 Len Lindsay
30 REM *            Idea from
40 REM *     CALCULATOR/COMPUTERS
50 DIM XX$(3)
60 GRAPHICS 3
70 REM FOLLOWING WILL BE USED AS COLOR R
EGISTER NUMBERS
80 LINE=1
90 CENTER=3
100 MARK=2
110 OFF=0
120 HINT=MARK
130 COLUMN=INT(RND(1)*34)+3
140 ROW=INT(RND(1)*16)+2
150 GOSUB 640:REM CORNERS
160 COLOR LINE
170 PLOT COLUMN-2,ROW-2
180 DRAWTO COLUMN+2,ROW-2
190 DRAWTO COLUMN+2,ROW+2
```

```
200 DRAWTO COLUMN-2,ROW+2
210 DRAWTO COLUMN-2,ROW-2
220 PRINT "[CLEAR][BELL]CAN YOU GUESS WH
ERE THE CENTER OF "
230 PRINT "THE RECTANGLE IS?"
240 PRINT "FIRST - WHAT COLUMN (3-36)";
250 GOSUB 680:REM INPUT ROUTINE
260 COLUMNGUESS=INT(XX+0.05)
270 IF COLUMNGUESS<3 OR COLUMNGUESS>36 T
HEN 220
280 COLOR HINT
290 PLOT COLUMNGUESS,0
300 PLOT COLUMNGUESS,19
310 PRINT "[CLEAR][BELL]NOW WHAT ROW IS
THE CENTER IN"
320 PRINT "(2-17) OR HIT OPTION TO REDO
COLUMN"
330 TRAP 310:POKE 764,255:REM SET TRAP A
ND CLEAR BUFFER
340 IF PEEK(53279)=3 THEN COLOR OFF:PLOT
 COLUMNGUESS,0:PLOT COLUMNGUESS,19:GOTO
220
350 IF PEEK(764)=255 THEN 340
360 GOSUB 710:REM INPUT ROUTINE
370 ROWGUESS=INT(XX+0.05)
380 IF ROWGUESS<2 OR ROWGUESS>17 THEN 31
0
390 COLOR HINT
400 PLOT 0,ROWGUESS
410 PLOT 39,ROWGUESS
420 COLOR CENTER
430 PLOT COLUMNGUESS,ROWGUESS
440 IF COLUMNGUESS=COLUMN AND ROWGUESS=R
OW THEN 520
450 PRINT "[CLEAR][BELL]OOPS - LOOK AT W
HERE YOUR GUESS WAS."
460 PRINT "COLUMN ";COLUMNGUESS;", ROW "
;ROWGUESS;" MISSED THE CENTER"
470 PRINT " HIT START TO TRY THE SAME ON
E AGAIN"
480 PRINT "OR FOR A NEW RECTANGLE HIT SE
LECT";
490 IF PEEK(53279)=6 THEN GOSUB 610:GOTO
 150
```

```
500 IF PEEK(53279)=5 THEN GOSUB 610:GOTO
 130
510 GOTO 490
520 RIGHT=RIGHT+1
530 PRINT "[CLEAR][BELL]YOU GOT IT - THA
T MAKES ";RIGHT;" RIGHT"
540 PRINT "HIT START TO TRY ANOTHER"
550 PRINT "HIT OPTION TO TURN HINT MARKS
 ";
560 IF HINT=OFF THEN PRINT "ON"
570 IF HINT=MARK THEN PRINT "OFF"
580 IF PEEK(53279)=6 THEN GOSUB 610:GOTO
 130
590 IF PEEK(53279)=3 THEN GOSUB 610:GOSU
B 630:GOTO 130
600 GOTO 580
610 GRAPHICS 3
620 RETURN
630 HINT=MARK+OFF-HINT:REM SWITCH
640 COLOR HINT
650 PLOT 0,0:PLOT 0,19:PLOT 39,0:PLOT 39
,19
660 RETURN
670 END
680 REM INPUT SUBROUTINE WITH BEEP
690 PRINT "[BELL]";
700 POKE 764,255
710 TRAP 710
720 INPUT XX$
730 TRAP 750
740 XX=VAL(XX$)
750 TRAP 34567
760 RETURN
```

הנפשה

הפקודות PLOT ו- DRAWTO יכולות לשמש אותך לא רק לציור תמונות, אלא גם לגרום לכך שהציורים ינועו על פני המסך, ישנו צורה ויהפכו לדמויות "חיות". למעשה, רוב הסרטים המצויירים כיום נעשים באמצעים ממוחשבים הדומים לאלה שנראה לך כאן.

זוהי תכנית הנפשה בסיסית. היא מזיזה נקודה על פני המסך. רצוי לנסות אותה לפני שממשיכים להנפשה יותר מסובכת.

```
10 REM *   <<< ANIMATION 1 >>>
20 REM *        moving dot
30 GRAPHICS 3
40 COLOR 2
50 FOR I=0 TO 39
60 GRAPHICS 3
70 PLOT I,10
80 GOSUB 110
90 NEXT I
100 END
110 FOR W=1 TO 300:NEXT W:RETURN
```



שימו לב כיצד פועלת התכנית. הלולאה המתחילה בשורה 50 מגדירה קודם כל את המצב הגרפי. אח"כ מציירת נקודה, עוברת לתת תכנית של השהיה ואז, לפני שהיא מציירת נקודה חדשה מעדכנת את המצב הגרפי וע"י כך מנקה את המסך. ע"י עדכון המצב הגרפי בכל פעם, מוחקים את הנקודה האחרונה שצוירה על המסך. זהו דבר שיוצר את האשליה של תנועת הנקודה על פני המסך. אם נסלק את שורה 60, כמו בתכנית למטה, נקבל שורה הגדלה לעינינו על המסך, ולא נקודה נעה.

```
10 GRAPHICS 3:COLOR 2
50 FOR I=1 TO 39
70 PLOT I,10
80 GOSUB 1000
90 NEXT I
100 END
1000 FOR W=1 TO 500:NEXT W:RETURN
```

אם רוצים להאיץ את תנועת הנקודה או להאט אותה, כל מה שצריך לעשות הוא לשנות את לולאת ההשהיה בשורה 1000. למשל :

```
1000 FOR W=1 TO 10:NEXT W:RETURN
```

היא מהירה, והלולאה

```
1000 FOR W=1 TO 5000:NEXT W:RETURN
```

מתנהלת לאטה.

כל מה שצריך לעשות כדי להזיז תמונה של פנים על פני המסך הוא כמה שנויים פשוטים בתכנית הראשונה. נשנה את שורה 50 כדי שנוכל לחזור ולצייר את גבולות הפנים

בצורה נאותה בנקודות חדשות על המסך. שורות 90-70 באות במקום שורה 70 בתכנית הקודמת כדי לכלול יותר מנקודה אחת (כלומר, את הפנים). כדאי לנסות את התכנית הזו, ובואו נראה מה קורה.

```
10 REM *    <<< ANIMATION 2 >>>
20 REM *     draws moving face
30 GRAPHICS 3
40 COLOR 2
50 FOR I=4 TO 35
60 GRAPHICS 3
70 PLOT I-2,8:PLOT I+2,8
80 PLOT I,10
90 PLOT I-1,12:PLOT I,12:PLOT I+1,12
100 GOSUB 130
110 NEXT I
120 END
130 FOR W=1 TO 300:NEXT W:RETURN
```

והנה התכנית, כשהיא מכילה את קו הגבול של הפנים :

```
10 REM *     <<< ANIMATION 3 >>>
20 REM *      face with border
30 FOR I=4 TO 74
40 GRAPHICS 5:COLOR 2
50 GOSUB 130
60 PLOT I-2,18:PLOT I+2,18
70 PLOT I,20
80 PLOT I-1,22:PLOT I,22:PLOT I+1,22
90 GOSUB 120
100 NEXT I
110 END
120 FOR W=1 TO 10:NEXT W:RETURN
130 REM  BORDER
140 PLOT I-4,15:DRAWTO I+4,15
150 DRAWTO I+4,26:DRAWTO I-4,26
160 DRAWTO I-4,15
170 RETURN
```

ועכשו משהו עוד יותר משוכלל, הכולל "התנגשות", תוך שמוש ב GRAPHICS 5 :

```
10 REM *    <<< ANIMATION 4 >>>
20 REM * face collides with square
30 REM * moving in opposite direction
40 GRAPHICS 5
50 COLOR 2
60 FOR I=4 TO 34
70 GRAPHICS 5
80 GOSUB 250:REM DRAW BORDER TO FACE
90 PLOT I-2,18:PLOT I+2,18
100 PLOT I,20
110 PLOT I-1,22:PLOT I,22:PLOT I+1,22
120 GOSUB 300:REM DRAW THE SQUARE FROM T
HE OTHER SIDE
130 GOSUB 230:REM PAUSE
140 NEXT I
150 FOR J=1 TO 25
160 SOUND 0,INT(RND(1)*256),INT(RND(1)*8
)*2,10
170 SETCOLOR 4,INT(RND(1)*16),INT(RND(1)
*8)*2
180 COLOR 3:PLOT 36,22:DRAWTO INT(RND(1)
*80),INT(RND(1)*40)
190 NEXT J
200 GRAPHICS 5:PRINT "THE REAL END"
210 END
220 REM *  PAUSE
230 FOR W=1 TO 10:NEXT W:RETURN
240 REM *  BORDER OF FACE
250 PLOT I-4,15:DRAWTO I+4,15
260 DRAWTO I+4,26:DRAWTO I-4,26
270 DRAWTO I-4,15
280 RETURN
290 REM *  BOX MOVING LEFT
300 PLOT 70-I,18:DRAWTO 76-I,18
310 DRAWTO 76-I,26:DRAWTO 70-I,26
320 DRAWTO 70-I,18
330 RETURN
```

הדוגמאות האלה יכולות לעזור לך להתחיל לפתח את הסרטים המצויירים שלך. אוצר המלים ב BASIC שהשתמשנו בו עד כה הוצג בחלק הראשון של הספר. וכעת נציג כמה דוגמאות יותר מסובכות של מה שאפשר לעשות עם התכונות המיוחדות של מחשב אטארי שלך.

הודעות נעות

התכניות הבאות מראות דרכים להציג הודעה נעה על המסך, תוך שמוש ב GRAPHICS 1 או GRAPHICS 2 (אותיות גדולות) של אטארי.

התכנית הראשונה מראה כמה דמויות מצויירות פשוטות, שאותן משיגים ע"י הצגה רציפה של קטעים של ההודעה. (שורה 190).

```
10 REM * <<< MESSAGE BILLBOARD >>>
20 REM *  (C) LEN LINDSAY, 1980
30 DIM MESSAGE$(200)
40 DIM SP$(200)
50 GRAPHICS 2
60 PRINT "ENTER MESSAGE, PLEASE"
70 INPUT MESSAGE$
80 SP$="                    "
90 SP$(LEN(SP$)+1)=MESSAGE$
100 MESSAGE$=SP$
110 MESSAGE$(LEN(MESSAGE$)+1)=" "
120 ML=LEN(MESSAGE$)
130 POKE 752,1
140 PRINT "[CLEAR]":REM CLEAR TEXT WINDO
W
150 SETCOLOR 2,0,0
160 FOR P=1 TO ML
170 EL=P+19:IF EL>ML THEN EL=ML
180 POSITION 0,5
190 PRINT #6;MESSAGE$(P,EL)
200 FOR W=1 TO 99:NEXT W
210 NEXT P
220 GOTO 160
```

כאשר משתמשים ב GRAPHICS 1 או ב GRAPHICS 2 מוכרחים לציין "#6;" אחרי המלה PRINT. (אחרי "#6;" יבואו המחרוזת או הטקסט המכיל את ההודעה). כפי שהסברנו קודם לכן, "#6" היא מילה שמורה, שה BASIC של אטארי מבין אותה ואשר מתיחסת להצגת טקסט מוגדל במסך הגרפי בשני המצבים המיוחדים האלה. הנה שתי דוגמאות :

```
4000 GRAPHICS 1
5000 PRINT #6;$MESSAGE
```

או

```
4000 GRAPHICS 2
5000 PRINT #6; "MY MESSAGE"
```

(כדאי לשים **לב** שבשתי הדוגמאות הללו יש מסך מפוצל ;
החצי העליון של המסך מיועד לטקסט באותיות בגודל רגיל.
הפקודה PRINT #6 מתיחסת רק לחצי העליון של המסך).

התכנית הבאה שנכתבה ע"י ביל קאריס מחברת אטארי, מראה
דרך פשוטה אבל משעשעת להצגת מחשב אטארי בפני חברים.
היא משתמשת בצרוף של אפקטים יפים של צבע ושל GRAPHICS 7
ואז מבזיקה את שמך (או את שם החברה או החבר) באותיות
גדולות. ויש גם קולות !

```
10 REM *       <<< GOODNAME >>>
20 REM *  By Bill Carris,  ATARI INC
30 PRINT "[CLEAR]"
40 DIM A$(9)
50 PRINT "ENTER NAME - UP TO 9 CHARACTER
S:"
60 INPUT A$
70 GRAPHICS 7+16
80 COLOR 1
90 Z=2:Y=5:X=155
100 PLOT X,Y
110 X=X+2*Z
120 SOUND 0,X,14,8
130 DRAWTO X,Y
140 Z=Z-(2*Z)
150 Y=Y+Z
160 DRAWTO X,Y
170 IF Z>0 THEN Z=Z+2
180 Z=Z-3
190 IF Y<70 THEN 110
200 FOR DD=1 TO 100:SOUND 3,DD,6,10
210 GRAPHICS 2+16:POSITION 3,8:PRINT #6;
"HELLO, ";A$:NEXT DD
220 FOR XYZ=1 TO 20:NEXT XYZ
230 GOTO 70
```

נתבונן רגע בשורות 70 ו-210 בתכנית של ביל.
שורה 70 :

70 GRAPHICS 7+16

ושורה 210 מתחילה :

210 GRAPHICS 2+16:

ע"י הוספת 16 למצבים הגרפיים בצורה כפי שזה נעשה כאן, מבטלים את "המסך המפוצל", ומקבלים מסך מלא לתצוגה של אותיות גדולות או צורות גרפיות. מידע נוסף לגבי השמוש בטכניקה הזו נמצא במדריך למשתמש של ATARI BASIC.

כמה ניסויים בהנדסה

אנו מציגים: אמנות דינמית מן המחשב

אנו מביאים כאן כמה תכניות שצריכות לתת לך תחושה לגבי האפשרויות הגרפיות של מחשב אטארי   נתחיל בכמה ריבועים :

```
10 REM *      <<< SQURVES >>>
20 REM *   (C) 1980 by Len Lindsay
30 REM * curved design made with lines
40 GRAPHICS 7+16
50 FOR X=0 TO 95 STEP 5
60 COLOR 1
70 PLOT 0,X:DRAWTO 1.5*X,95
80 COLOR 2
90 DRAWTO 143,95-X
100 COLOR 3
110 DRAWTO 143-1.5*X,0
120 COLOR 2
130 DRAWTO 0,X
140 NEXT X
150 GOTO 150
```

ועכשו כמה עגולים ועכבישים, שהם שמוש יותר מורכב של צורות מעגליות :

עיגולים:

```
10 REM *      <<< CIRCLES >>>
20 REM *  (C) 1980 by Len Lindsay
30 TRAP 210
40 GRAPHICS 7+16:LUM=12
50 CENTERX=RND(1)*160
60 CENTERY=RND(1)*96
```

```
70 DISTANCE=RND(1)*25+25
80 FOR SHAPEX=0 TO DISTANCE
90 LUM=18-LUM:SETCOLOR 2,5,LUM
100 SHAPEY=SQR(DISTANCE*DISTANCE-SHAPEX*
SHAPEX)
110 FOR DIF=0.25 TO 0.75 STEP 0.25
120 COLOR DIF*4
130 PLOT CENTERX+SHAPEX*DIF,CENTERY+SHAP
EY*DIF
140 PLOT CENTERX-SHAPEX*DIF,CENTERY+SHAP
EY*DIF
150 PLOT CENTERX+SHAPEX*DIF,CENTERY-SHAP
EY*DIF
160 PLOT CENTERX-SHAPEX*DIF,CENTERY-SHAP
EY*DIF
170 NEXT DIF
180 NEXT SHAPEX
190 GOTO 50
200 END
210 TRAP 210:GOTO 10+PEEK(186)+PEEK(187)*256
```

ועכבישים:

```
10 REM *        <<< SPIDERS >>>
20 REM *   a variation on circles
30 REM *   (C) 1980 by Len Lindsay
40 TRAP 220
50 GRAPHICS 7+16:LUM=12
60 CENTERX=RND(1)*160
70 CENTERY=RND(1)*96
80 DISTANCE=RND(1)*25+25
90 FOR SHAPEX=0 TO DISTANCE
100 LUM=18-LUM:SETCOLOR 2,5,LUM
110 SHAPEY=DISTANCE-SQR(DISTANCE*DISTANC
E-SHAPEX*SHAPEX)
120 FOR DIF=0.25 TO 0.75 STEP 0.25
130 COLOR DIF*4
140 PLOT CENTERX+SHAPEX*DIF,CENTERY+SHAP
EY*DIF
150 PLOT CENTERX-SHAPEX*DIF,CENTERY+SHAP
EY*DIF
160 PLOT CENTERX+SHAPEX*DIF,CENTERY-SHAP
EY*DIF
170 PLOT CENTERX-SHAPEX*DIF,CENTERY-SHAP
EY*DIF
```

```
180 NEXT DIF
190 NEXT SHAPEX
200 GOTO 60
210 END
220 TRAP 220:GOTO 10+PEEK(186)+PEEK(187)
*256
```

ועתה, כמה יהלומים מנצנצים בגדלים ובצבעים שונים :

```
10 REM *      <<< DIAMONDS >>>
20 REM * (C) 1980 by Len Lindsay
30 LUM=12
40 GRAPHICS 7+16
50 TRAP 220:REM FOR OUT OF RANGE
60 CENTERX=RND(1)*160
70 CENTERY=RND(1)*96
80 DISTANCE=RND(1)*25+25
90 FOR SHAPEX=0 TO DISTANCE
100 LUM=18-LUM:SETCOLOR 2,5,LUM
110 SHAPEY=DISTANCE-SHAPEX
120 FOR DIF=0.25 TO 0.75 STEP 0.25
130 COLOR DIF*4
140 PLOT CENTERX+SHAPEX*DIF,CENTERY+SHAP
EY*DIF
150 PLOT CENTERX-SHAPEX*DIF,CENTERY+SHAP
EY*DIF
160 PLOT CENTERX+SHAPEX*DIF,CENTERY-SHAP
EY*DIF
170 PLOT CENTERX-SHAPEX*DIF,CENTERY-SHAP
EY*DIF
180 NEXT DIF
190 NEXT SHAPEX
200 GOTO 60
210 END
220 TRAP 220:GOTO 10+PEEK(186)+PEEK(187)
*256
```

ובתכנית זו יראה הכל משולש. לא לדאוג, אלה לא העינים שלך.

```
10 REM *          <<< LINES >>>
20 REM *        by Len Lindsay
30 DIM X(3),Y(3)
40 GRAPHICS 7+16
50 FOR R=0 TO 3
60 X(R)=INT(RND(1)*160)
70 Y(R)=INT(RND(1)*96)
80 NEXT R
90 FOR R=0 TO 3
110 COLOR R
120 SETCOLOR 2,RND(1)*256,4:REM THIS
  COMMAND CHANGES THE LINE COLORS
130 PLOT X(R),Y(R)
140 DRAWTO X,Y
150 COLOR 0:PLOT X(R),Y(R)
160 NEXT R
170 GOTO 90
```

וכתוספת לתכנית הקודמת, קצת הבהוב :

```
5 REM LINES2 (RANDOM LINES WITH CHANGING COLORS,
  GIVING FLASHING EFFECTS)
6 REM BY LEN LINDSAY
10 DIM X(3),Y(3)
20 GRAPHICS 7+16
100 FOR R=0 TO 3
110 X(R)=INT(RND(1)*160)
120 Y(R)=INT(RND(1)*96)
130 NEXT R
200 FOR R=0 TO 3
210 X=INT(RND(1)*160):Y=INT(RND(1)*96)
220 COLOR R
225 SETCOLOR 2,RND(1)*256,4:REM THIS COMMAND
    CHANGES THE LINE COLORS
230 PLOT X(R),Y(R)
240 DRAWTO X,Y
245 COLOR 0:PLOT X(R),Y(R)
250 NEXT R
300 GOTO 200
```

ולבסוף, צורה גרפית להנאתך :

```
10 REM *      <<< SPHERES >>>
20 REM *  (C) 1980 by Len Lindsay
30 TRAP 210
40 GRAPHICS 7+16:LUM=12
50 CENTERX=RND(1)*160
60 CENTERY=RND(1)*96
70 DISTANCE=RND(1)*25+25
80 FOR SHAPEX=0 TO DISTANCE
90 LUM=18-LUM:SETCOLOR 2,5,LUM
100 SHAPEY=SQR(DISTANCE*DISTANCE-SHAPEX*
SHAPEX)
110 FOR DIF=0.25 TO 0.75 STEP 0.25
120 COLOR DIF*4
130 PLOT CENTERX+SHAPEX,CENTERY+SHAPEY*D
IF
140 PLOT CENTERX-SHAPEX,CENTERY+SHAPEY*D
IF
150 PLOT CENTERX+SHAPEX,CENTERY-SHAPEY*D
IF
160 PLOT CENTERX-SHAPEX,CENTERY-SHAPEY*D
IF
170 NEXT DIF
180 NEXT SHAPEX
190 GOTO 50
200 END
210 TRAP 210:GOTO 10+PEEK(186)+PEEK(187)*256
```

שורות ישרות,או:
מתימטיקה היא יותר ממה שנראה לעין

התכנית הבאה היא תכנית הדגמה מקובלת באטארי. היא נכתבה ע"י מתכנתים של אטארי זמן קצר אחרי שהצגנו לראשונה את המחשבים האישיים שלנו, והיא מראה מה אפשר לעשות באמצעות שורות ישרות בלבד ! מי אמר שמתמטיקה צריכה להיות משעממת ? הקטע האחרון מענין במיוחד וכדאי להתבונן בו בתשומת לב מיוחדת, משום שהוא מציג טריקים של צבע ותנועה המטעים את עינינו. (חשוב : לתכנית זו דרוש אטארי 400 או 800 עם מינימום 16K זכרון).

```
10 REM *          <<< PRETTY >>> △
20 REM *     Graphics demonstrations
30 REM *   by ATARI Software Engineers
40 GOSUB 1000
50 C=0:Q=1:SETCOLOR 1,5,5
60 DEG
70 XI=80:YI=50
80 GRAPHICS 7+16
90 PLOT XI,YI
100 FOR I=1 TO 1000 STEP 5
110 Q=Q+1
120 IF Q>3.5 THEN Q=1
130 COLOR Q
140 R=I/10
150 T=I
160 X=R*COS(T)
170 Y=R*SIN(T)
180 IF Y+YI<0 THEN 240
190 PLOT X+XI,Y+YI
200 X=(I+C)/16*COS(I+C+90)
210 Y=(I+C)/16*SIN(I+C+90)
220 DRAWTO X+XI,Y+YI
230 NEXT I
240 SETCOLOR 2,8,2
250 SETCOLOR 1,8,5
260 SETCOLOR 0,8,8
270 GOSUB 370
280 SETCOLOR 0,8,2
290 SETCOLOR 2,8,5
300 SETCOLOR 1,8,8
310 GOSUB 370
320 SETCOLOR 1,8,2
330 SETCOLOR 0,8,5
340 SETCOLOR 2,8,8
350 GOSUB 370
360 GOTO 240
370 FOR K=1 TO 13:NEXT K
380 RETURN
1000 REM ______AL'S DEMO__________
```

```
1010 DEG
1020 GRAPHICS 8+16
1030 COLOR 1
1040 SETCOLOR 2,0,0
1050 FOR I=1 TO 360 STEP 5
1060 X=319*I/360
1070 Y=80+80*SIN(I)
1080 IF I>270 THEN 1100
1090 PLOT 0,0
1100 DRAWTO X,Y
1110 IF I<90 THEN 1130
1120 DRAWTO 319,159
1130 NEXT I
1140 GOSUB 6000
2000 REM __________DBLDOM______________
2010 DEG
2020 A=INT(1.9*160)
2030 GRAPHICS 8+16
2040 SETCOLOR 2,0,0
2050 FOR I=0 TO 160 STEP 5
2060 B=INT(I/2)
2070 COLOR 1
2080 PLOT 0,B
2090 DRAWTO I,160
2100 PLOT A,B
2110 DRAWTO A-I,160
2120 PLOT 0,160-B
2130 DRAWTO I,0
2140 PLOT A,160-B
2150 DRAWTO A-I,0
2160 NEXT I
2170 GOSUB 6000
3000 REM ________RANDOM BOXES__________
3010 GRAPHICS 7+16
3020 FOR I=1 TO 32
3030 X1=INT(RND(0)*160)
3040 Y1=INT(RND(0)*96)
3050 X2=INT(RND(0)*160)
3060 Y2=INT(RND(0)*96)
3070 IF X1=X2 OR Y1=Y2 THEN 3030
3080 COLOR INT(RND(0)*3+1)
3090 PLOT X1,Y1
3100 DRAWTO X1,Y2
3110 DRAWTO X2,Y2
```

```
3120 DRAWTO X2,Y1
3130 DRAWTO X1,Y1
3140 NEXT I
3150 GOSUB 6000
4000 REM ____COLLAPSING BOXES_____
4010 TRAP 4170:J=0
4020 GRAPHICS 7+16
4030 X1=INT(RND(0)*80)
4040 Y1=INT(RND(0)*48)
4050 X2=X1+INT(RND(0)*80)
4060 Y2=Y1+INT(RND(0)*48)
4070 IF X1=X2 OR Y1=Y2 THEN 3030
4080 COLOR INT(RND(0)*3+1)
4090 PLOT X1,Y1
4100 DRAWTO X1,Y2
4110 DRAWTO X2,Y2
4120 DRAWTO X2,Y1
4130 DRAWTO X1,Y1
4140 X1=X1+2:Y1=Y1+2
4150 X2=X2-1:Y2=Y2-1
4160 GOTO 4080
4170 J=J+1:TRAP 4170
4180 GOSUB 6000
4190 IF J<4 THEN 4020
4200 TRAP 40000
5000 REM _________MOIRE_________
5010 MX=320:MY=192:MODE=8
5020 MX=MX-1:MY=MY-1
5030 GRAPHICS MODE+16
5040 IF MODE=8 THEN SETCOLOR 2,0,0
5050 FOR Y=MY TO 0 STEP -1
5060 IF PEEK(764)<>255 THEN RETURN
5070 Q=Y:GOSUB 5300
5080 PLOT 0,Y
5090 DRAWTO MX/2,MY
5100 NEXT Y
5110 FOR X=0 TO MX
5120 IF PEEK(764)<>255 THEN RETURN
5130 Q=X:GOSUB 5300
5140 PLOT X,0
5150 DRAWTO MX/2,MY
5160 NEXT X
5170 FOR Y=0 TO MY
5180 IF PEEK(764)<>255 THEN RETURN
```

```
5190 Q=Y:GOSUB 5300
5200 PLOT MX,Y
5210 DRAWTO MX/2+1,MY
5220 NEXT Y
5230 RETURN
5240 FOR X=MX TO 0 STEP -1
5250 Q=X:GOSUB 5300
5260 PLOT X,MY
5270 DRAWTO MX/2,MY/2
5280 NEXT X
5290 RETURN
5300 IF Q=125 THEN Q=13
5310 IF Q=155 THEN Q=11
5320 COLOR Q
5330 RETURN
5340 END
6000 FOR K=1 TO 500
6010 NEXT K:RETURN
```

ציור עם דרגת הפרדה גבוהה

התכנית הבאה היא דוגמה ליכולת הגרפית המצויינת של מחשב אטארי. כשהוא משתמש בדרגת ההפרדה הגבוהה ביותר שלו(GRAPHICS 8). התכנית נכתבה בידי טנדי טרואר, ומציירת תמונה של מחשב אטארי 400 ! (גם תכנית זו דורשת מינימום של 16K זכרון).

```
10 REM *  <<< ATARI 400 DRAWING >>>
20 REM *   by Tandy Trower (ATARI)
30 DIM Y$(1)
40 PRINT "[CLEAR]WOULD YOU BELIEVE THAT
THE ATARI ":PRINT "COMPUTER CAN DRAW ITS
 OWN":? "ATARI 400 PORTRAIT";
50 INPUT Y$
60 IF Y$="N" THEN PRINT :PRINT "WELL, YO
U'RE IN FOR A SURPRISE..."
70 IF Y$<>"N" THEN PRINT :PRINT "WATCH A
N ARTISTIC GENIUS AT WORK..."
80 FOR WAIT=1 TO 1400:NEXT WAIT
90 DIM A$(10)
100 GRAPHICS 8
```

```
110 SETCOLOR 4,12,4
120 POKE 752,1
130 SETCOLOR 2,0,14:SETCOLOR 1,0,2
140 DL=PEEK(560)+PEEK(561)*256
150 D1=PEEK(DL+4)+PEEK(DL+5)*256
160 PX=16:PY=43
170 A$="ATAR 400"
180 FOR U=1 TO LEN(A$):D2=57344+((ASC(A$
(U,U))-32)*8):D3=D1+PY*40+PX+U-1:FOR Z=0
 TO 7:POKE D3+Z*40,PEEK(D2+Z)
190 NEXT Z:NEXT U
200 COLOR 3:PLOT 162,44:DRAWTO 162,49:PL
OT 163,44:DRAWTO 163,49
210 COLOR 1
220 FOR I=1 TO 7:PLOT 117,42+I:DRAWTO 12
5,42+I:NEXT I
230 COLOR 2:PLOT 118,48:DRAWTO 121,45:DR
AWTO 124,48
240 PLOT 121,44:DRAWTO 121,48
250 COLOR 1:PLOT 112,40:DRAWTO 195,40:DR
AWTO 195,53:DRAWTO 112,53:DRAWTO 112,40
260 REM LARGE BOX
270 PLOT 86,17:DRAWTO 112,17:PLOT 200,17
:DRAWTO 226,17
280 DRAWTO 250,76:DRAWTO 62,76:DRAWTO 86
,17
290 PLOT 79,38:DRAWTO 108,38:PLOT 204,38
:DRAWTO 233,38
300 PLOT 86,17:DRAWTO 79,38:DRAWTO 62,76
310 PLOT 226,17:DRAWTO 233,38:DRAWTO 250
,76
320 PLOT 110,27:DRAWTO 201,27
330 DRAWTO 204,39:DRAWTO 204,68
340 DRAWTO 171,68:DRAWTO 171,62:DRAWTO 1
39,62:DRAWTO 139,68
350 DRAWTO 107,68:DRAWTO 107,39:DRAWTO 1
10,27
360 PLOT 139,68:DRAWTO 139,72:DRAWTO 171
,72:DRAWTO 171,68
370 Y=63:FOR I=0 TO 9 STEP 2:PLOT 140,Y+
I:DRAWTO 170,Y+I:NEXT I
380 X=112:FOR I=1 TO 90 STEP 3:PLOT X+I,
17:DRAWTO X+I,23:NEXT I
390 PLOT 81,27:DRAWTO 56,27
```

```
400 DRAWTO 28,122
410 DRAWTO 44,122:DRAWTO 46,125:DRAWTO 2
66,125:DRAWTO 268,122
420 DRAWTO 284,122
430 DRAWTO 256,27:DRAWTO 231,27
440 PLOT 28,122:DRAWTO 27,129
450 DRAWTO 82,129:DRAWTO 89,142:DRAWTO 2
21,142:DRAWTO 228,129
460 DRAWTO 285,129:DRAWTO 284,122
470 PLOT 27,129:DRAWTO 28,135
480 DRAWTO 82,135:DRAWTO 87,144:DRAWTO 2
23,144:DRAWTO 228,135
490 DRAWTO 283,135:DRAWTO 284,130
500 PLOT 85,132:DRAWTO 225,132
510 PLOT 53,80:DRAWTO 259,80
520 DRAWTO 268,122
530 PLOT 53,80:DRAWTO 44,122
540 PLOT 236,83:DRAWTO 244,122
550 PLOT 57,83:DRAWTO 236,83
560 PLOT 56,83:POSITION 47,122:POKE 765,
1:XIO 18,#6,0,0,"S:"
570 PLOT 53,135:DRAWTO 57,142
580 DRAWTO 85,142
590 PLOT 259,135:DRAWTO 255,142:DRAWTO 2
24,142
600 POKE 765,1:PLOT 84,132:POSITION 88,1
41:XIO 18,#6,0,0,"S:"
610 REM SELECT/OPTION ETC.
620 PLOT 240,83:DRAWTO 248,122
630 PLOT 239,82:PLOT 247,123
640 PLOT 244,83:DRAWTO 252,122
650 PLOT 256,83:DRAWTO 264,122
660 PLOT 245,83:DRAWTO 256,83
670 PLOT 246,89:DRAWTO 257,89
680 PLOT 246,91:DRAWTO 258,91
690 PLOT 247,92:DRAWTO 258,92
700 PLOT 248,97:DRAWTO 259,97
710 PLOT 250,107:DRAWTO 261,107
720 PLOT 250,108:DRAWTO 261,108
730 PLOT 251,114:DRAWTO 263,114
740 PLOT 252,116:DRAWTO 263,116
750 PLOT 252,117:DRAWTO 263,117
760 PLOT 253,122:DRAWTO 264,122
770 PLOT 252,100:DRAWTO 255,100
```

```
780 PLOT 251,101:DRAWTO 256,101
790 PLOT 251,102:DRAWTO 257,102
800 PLOT 252,103:DRAWTO 256,103
810 COLOR 4
820 PLOT 253,101:DRAWTO 254,101
830 X=59:Y=86:NK=15:L=6:GOSUB 1040
840 NC=9
850 X=71:Y=86:NK=13:L=1:GOSUB 1070
860 X=75:Y=93:NK=12:L=6:GOSUB 1040
870 X=183:Y=93:NK=2:L=1:GOSUB 1070
880 X=77:Y=100:NK=13:L=6:GOSUB 1040
890 X=185:Y=100:NK=4:L=1:GOSUB 1070
900 X=83:Y=107:NK=11:L=6:GOSUB 1040
910 X=167:Y=107:NK=3:L=1:GOSUB 1070
920 X=89:Y=114:NK=1:L=106:GOSUB 1040
930 X=57:Y=93:NK=1:L=10:GOSUB 1040
940 NC=13:X=57:Y=93:NK=1:L=1:GOSUB 1070
950 X=221:Y=93:NK=1:L=12:GOSUB 1040
960 X=59:Y=100:NK=1:L=10:GOSUB 1040
970 X=59:L=1:FL=1:GOSUB 1070
980 X=59:Y=107:NK=1:L=16:GOSUB 1040
990 X=59:L=1:NC=19:GOSUB 1070
1000 X=217:Y=107:NK=1:L=10:GOSUB 1040:X=
217:L=1:NC=13:GOSUB 1070
1010 ? "   [Q][R][R][R][R][R][R][R][R][R
][R][R][R][R][R][R][R][R][R][R][R][R][R]
[R][R][R][R][R][R][E]":? "    | ATARI PE
RSONAL COMPUTERS  |"
1020 ? "   [Z][R][R][R][R][R][R][R][R][R
][R][R][R][R][R][R][R][R][R][R][R][R][R]
[R][R][R][R][R][R][C]";
1030 GOTO 1030
1040 FOR I=1 TO NK:PLOT X,Y:DRAWTO X+L,Y
:DRAWTO X+L+1,Y+1:DRAWTO X+L+1,Y+4:DRAWT
O X+L,Y+5:DRAWTO X,Y+5
1050 DRAWTO X-1,Y+4:DRAWTO X-1,Y+1:DRAWT
O X,Y:X=X+12:NEXT I
1060 RETURN
1070 FOR I=1 TO NK
1080 FOR K=1 TO NC:COLOR K:PLOT X,Y+1:DR
AWTO X+L,Y+1:PLOT X,Y+2:DRAWTO X+L,Y+2:I
F FL=1 THEN 1100
1090 GOTO 1110
1100 PLOT X,Y+3:DRAWTO X+L,Y+3:PLOT X,Y+
```

```
4:DRAWTO X+L,Y+4
1110 X=X+1:NEXT K:X=X+3:NEXT I:COLOR 4:RETURN
```

גרפיקה של טקסט ושל סימנים גרפיים

שתי התכניות האחרונות בפרק זה מלמדות כיצד אפשר ליצור גרפיקה ודמויות מצוירות ונעות בלי אפילו להשתמש במצבים הגרפיים המיוחדים של מחשב אטארי. שתי התכניות משתמשות ב GRAPHICS 0, שהוא המצב הגרפי הרגיל לטקסט ומשמש אותך תמיד לכתיבת התכניות שלך ב BASIC.

התכנית הראשונה מראה כיצד אפשר להשיג אפקט מעניין של גלים תוך שמוש באותיות של המלה. אפשר להשתמש בשמך הפרטי, או במלה שמשמעותה קשורה בתנועה או בקצב. צריך לנסות, ולתת לדמיון להשתולל !

```
10 REM *   <<< SINEWAVE SCREEN >>>
20 REM *    (C) 1980 LEN LINDSAY
30 DIM XX$(10)
40 POKE 752,1
50 PRINT "[CLEAR]TYPE IN A WORD - UP TO
10 CHARACTERS"
60 GOSUB 180
70 L=6
80 FOR Z=0 TO 99 STEP 0.3
90 X=20-LEN(XX$)/2+(13*SIN(Z))
100 FOR LOOP=1 TO X
110 PRINT " ";
120 NEXT LOOP
130 PRINT XX$
140 L=L+0.1:IF L=19 THEN L=6
150 SETCOLOR 1,0,L
160 NEXT Z
170 RUN
180 PRINT "[BELL][BELL]";
190 POKE 764,255
200 TRAP 200
210 INPUT XX$
220 TRAP 240
230 XX=VAL(XX$)
240 TRAP 34567
250 RETURN
```

התכנית השניה, שנכתבה ע"י טנדי טרואר, משתמשת בסימנים גרפיים (באמצעות (CTRL) עם אותיות על לוח המקשים), באפקטים קוליים (לא לשכוח להגביר את הקול במכשיר הטלביזיה) ובהנפשה משוכללת כדי ליצור את מה שניתן לכנות "הגדרה מילונית דינמית", כלומר, הגדרה שבאמת עושה את מה שהיא אומרת ! צריך להעתיק את התכנית הזו בזהירות, ולזכור שאותיות בסוגריים מרובעים פירושן ללחוץ ולהחזיק את (CTRL) יחד עם הלחיצה על האות, וכן שבאותיות מודגשות בקו חייבים קודם ללחוץ על המקש עם סמל אטארי.

```
10 REM *   <<< HELICOPTER >>>
20 REM * Demo of animation & graphics
30 REM * with basic (no GR. commands)
40 REM * by Tandy Trower (1980)
50 GRAPHICS 0:SETCOLOR 2,9,12:SETCOLOR 1
,0,0:SETCOLOR 4,0,6
60 S=50
70 DIM A$(20),B$(20),C$(20),D$(20),E$(20
)
80 POKE 752,1:PRINT "[CLEAR]"
90 X=13:Y=19
100 POSITION 2,2:PRINT "HELICOPTER: an a
ircraft whose support":PRINT " in the ai
r is derived from the "
110 PRINT " reaction of a stream of air
driven":PRINT " downward by one or more
lifting "
120 PRINT " rotors turning about substan
tially":PRINT " vertical axes."
130 A$=" [R][R][R][R][R][R][S][R][R][R][
R][R][R]"
140 B$="     [F][M][M]_[J][N][N][N][N][F]
"
150 C$="     [V] [C]_[I][M][M][M][M][G]"
160 D$="    [B][M][M][M]|"
170 E$="    [M][M][M][M][M][M]"
180 POSITION X,Y:PRINT A$:POSITION X,Y+1
:PRINT B$:POSITION X,Y+2:PRINT C$:POSITI
ON X,Y+3:PRINT D$
190 POSITION X,Y+4:PRINT E$;
200 FOR T=0 TO 1000:NEXT T
210 FOR T=0 TO 300:NEXT T
```

1 בשורה 300, (DEL LINE) פרושו : ללחוץ על (ESC) ואז על (DEL) (SHIFT) ביחד.

```
220 R=20:S=240:M=1
230 FOR I=0 TO R:GOSUB 410:A$="        [S
]       ":POSITION X,Y:? A$
240 FOR D=0 TO S-(I*12):NEXT D
250 SOUND 1,220,6,M
260 A$=" [R][R][R][R][R][R]+[R][R][R][R]
[R][R]":POSITION X,Y:? A$:GOSUB 410:M=M+
0.1:NEXT I
270 R=70:GOSUB 350
280 R=0
290 FOR K=0 TO 23
300 POSITION 2,0:PRINT "[DEL LINE]":Y=Y-
1:FOR B=0 TO 4:NEXT B:GOSUB 350
310 IF K=10 OR K=18 THEN R=40:GOTO 330
320 R=0
330 NEXT K
340 GOTO 400
350 IF Y<0 THEN GOSUB 410:RETURN
360 FOR I=0 TO R:GOSUB 410:A$="        [S
]       ":POSITION X,Y:? A$
370 SOUND 1,220,6,6
380 A$=" [R][R][R][R][R][R]+[R][R][R][R]
[R][R]":POSITION X,Y:? A$:GOSUB 410
390 NEXT I:RETURN
400 SOUND 1,0,0,0:FOR B=0 TO 500:NEXT B:
GOTO 90
410 SOUND 0.10,0,14:SOUND 0,0,0,0:RETURN
```

פרק ז'



צלילי המוסיקה

פקודת SOUND (צליל) של ה BASIC של אטארי היא בעלת ארבעה משתנים : SOUND X, Y, Z, W
המשתנה W מאפשר לך לבחור אחד מתוך ארבעה קולות. תחום ה X הוא בין 0 ל-3, ולכל קול דרוש משפט SOUND נפרד.



המשתנה Y מאפשר לך לשלוט ב-255 גבהי צליל שונים. תחום המשתנה Y נע בין 1 ו-255, וככל שערכו של המשתנה גבוה יותר, כך נמוך יותר גובה הצליל המופק. ערכו של ה-דו האמצעי הוא 121. זוהי טבלה של ערכי גובה הצליל בסולם התוים המקובל :

טבלה לערכי גובה הצליל בסולם המוסיקלי

צלילים גבוהים	29	דו
	31	סי
	33	סי במול - לה דיאז
	35	לה
	37	לה במול - סול דיאז
	40	סול
	42	סול במול - פה דיאז
	45	פה
	47	מי
	50	מי במול - רה דיאז
	53	רה
	57	רה במול - דו דיאז
	60	לה
	64	סי
	68	סי במול - לה דיאז
	72	לה
	76	לה במול - סול דיאז
	81	סול
	85	סול במול - פה דיאז
	91	פה
	96	מי
	102	מי במול - רה דיאז
	108	רה
	114	רה במול - דו דיאז

דו אמצעי	121	דו
	128	סי
	136	סי במול - לה דיאז
	144	לה
	153	לה במול - סול דיאז
	162	סול
	173	סול במול - פה דיאז
	182	פה
צלילים נמוכים	193	מי
	204	מי במול - רה דיאז
	217	רה
	230	רה במול - דו דיאז
	243	דו

המשתנה Z שולט על גווני צליל ומאפשר לך ליצור אפקטים קוליים, כפי שידגימו כמה מהתכניות בפרק זה. התחום של Z הוא מ-0 עד 14. 10 יתן צליל נקי, שבו עליך להשתמש בכתיבת מוסיקה או בתכניות אחרות של צליל ללא אפקטים נלוים. הפיקוח על גווני הצליל נעשה רק באמצעות מספרים זוגיים. ההתנסות בהם תעזור לך לפתח אפקטים מיוחדים, לווי למשחקים ופסי צליל ללוות את הסרטים המצויירים שלך.

המשתנה הרביעי, W , מפקח על עצמת הצליל והטווח שלו הוא מ-1, שבקושי נשמע ועד 15 - צליל בעל עצמה רבה. 8 נחשב להגדרה סבירה של עצמת הצליל לרוב תכניות הקול. כדי להשתיק את הקול לגמרי, הפקודה היא SOUND X ,0 ,0 ,0.

והנה סכום של תחומי המספרים עבור כל אחד ממשתני הקול :

	עצמה	גוון	גובה	קול
צליל	W	Z	Y	X
	1-15 0 עבור השאר, או לניתוק	0-14 מספרים זוגיים	1-255	0-3

ועכשו הבה נשחק. התכניות הראשונות הפשוטות עוסקות בגובה, המשתנה השני.

גובה הצליל

אלה הם כמה צלילים אקראיים שיתנו לך תחושה של טווח הפעולה הקולית של האטארי שלך.

```
5 REM RANDOM SOUND
10 SOUND 0, RND(1)*255, 10, 12
20 FOR X=1 TO 200:NEXT X
30 GOTO 10
```

התכנית הבאה היא יותר מורכבת ותדגים אפקטים קוליים שאפשר להפיק מן האטארי. התכנית תעבור באופן מחזורי דרך הצלילים השונים של האטארי, תוך שמוש בפקודת SOUND. חשוב לזכור שאפשר ליצור איכויות צליל רבות ושונות נוספות, ע"י משחק במשך הצליל ובעצמתו, וכן ע"י צרוף כל ארבעת הקולות יחד-כל אחד עם צלילו המיוחד.

<u>לתשומת לבך</u> : בפקודת SOUND של אטארי, ככל שהמספר עבור גובה הצליל (Pitch) גבוה יותר, כך נמוכה יותר התדירות האמיתית של הצליל המופק. "גבהים" אלה נעים בין 255 (הנמוך ביותר) ל-1 (הגבוה ביותר) ; ו-0 משמש לניתוק מוחלט של צליל מסוים.

```
10 REM <<< ATARI SOUND TEST >>>
20 REM *  Set speed manually
30 REM *  by Ted Kahn, 1980
40 TONE=0
50 PRINT "[CLEAR]":POSITION 14,0:? "ATAR
I SOUNDS"
60 ? :? "THIS PROGRAM RUNS THROUGH SOME
OF THE":? "SOUNDS WHICH YOUR ATARI CAN M
AKE."
70 ? "THE PROGRAM WILL LET YOU CHOOSE A
":? "PAUSE, SO THAT YOU CAN HEAR THE RAN
GE"
80 ? "OF SOUNDS SLOWLY OR QUICKLY.":? :?
 :? :PAUSE=2000:GOSUB 280
90 GOSUB 240
100 POSITION 2,12
110 PRINT "PITCH","TONE"
120 POKE 752,1:REM TURN CURSOR OFF
130 FOR PITCH=255 TO 0 STEP -1
```

```
140 POSITION 2,14:PRINT PITCH,"    "
150 POSITION 12,14:PRINT TONE
160 SOUND 0,PITCH,TONE,8
170 GOSUB 280
180 NEXT PITCH
190 GOSUB 280
200 SOUND 0,0,0,0:POKE 752,0
210 TONE=TONE+2:IF TONE<=16 THEN 90
220 POKE 752,0:REM CURSOR BACK ON
230 END
240 PRINT "[CLEAR]":POSITION 2,8:PRINT "
SET PAUSE BETWEEN 0-1000"
250 PRINT "(0=NO PAUSE, 1000=LONG PAUSE)
";:INPUT PAUSE
260 PAUSE=INT(PAUSE):IF PAUSE<0 OR PAUSE
>1000 THEN 240
270 RETURN
280 REM PAUSE BETWEEN NOTES & TONES
290 FOR WAIT=0 TO PAUSE:NEXT WAIT
300 RETURN
```

ועכשו נעבור לכמה גרסאות מעוטרות של תכנית הקול הראשונה, עם כמה אפקטים חזותיים כתוספת :

הנקודה המרקדת

```
5 REM RANDOM SOUND
6 REM WITH DOT JUMPING AROUND THE SCREEN
10 SOUND 0,RND(1)*255, 10, 12
20 FOR X=1 TO 200:NEXT X
25 GRAPHICS 7:COLOR 1
26 PLOT RND(1)*158,RND(1)*78
30 GOTO 10
```

קווים במשבר

```
5 REM RANDOM SOUND
6 REM WITH LINES ALL AROUND
10 SOUND 0,RND(1)*255, 10, 12
20 FOR X=1 TO 200:NEXT X
25 GRAPHICS 7:COLOR 1
26 PLOT RND(1)*158,RND(1)*78
27 DRAWTO RND(1)*155,RND(1)*78
30 GOTO 10
```

מה יקרה אם נשנה את שורה 25 כך שתהיה שורה 8 ?

דוגמה לקו/קול אקראיים

(ע״י היפוך הלולאה תשארנה השורות על המסך)

```
5 REM RANDOM SOUND
6 REM WITH LINES ALL AROUND
7 REM SPEEDED UP
8 REM BY CHANGING AROUND THE LOOP
9 REM THE LINES STAY ON THE SCREEN
10 GRAPHICS 7:COLOR 1
20 FOR X=1 TO 20:NEXT X
25 SOUND 0,RND(1)*255, 10, 12
26 PLOT RND(1)*158,RND(1)*78
27 DRAWTO RND(1)*155,RND(1)*78
30 GOTO 25
```

צופר או משרוקית

כמה אפקטים פשוטים :

```
10 REM *      <<< SOUND 1 >>>
20 FOR I=255 TO 1 STEP -4
30 SOUND 0,I,10,10
40 GOSUB 70
50 NEXT I
60 END
70 FOR W=1 TO 100:NEXT W:RETURN
```

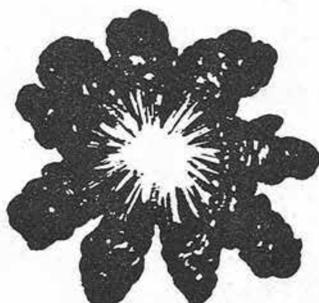
שיגור של חללית



```
10 REM *  <<< SOUND PROGRAM 2 >>>
20 REM *       ROCKET LIFT OFF
30 FOR I=255 TO 1 STEP -4
40 SOUND 0,I,8,10
50 GOSUB 80
60 NEXT I
70 END
80 FOR W=1 TO 50:NEXT W:RETURN
```

התפוצצות

ההפך משיגור



```
10 REM *       <<< SOUND 3 >>>
20 REM * explosion (reverse rocket)
30 FOR I=1 TO 255 STEP 4
40 SOUND 0,I,8,10
50 GOSUB 80
60 NEXT I
70 END
80 FOR W=1 TO 5:NEXT W:RETURN
```

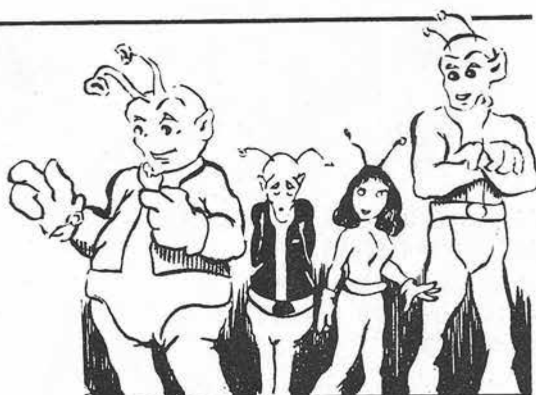
ועכשו התפוצצות, מלווה במנגינה פשוטה

```
10 REM *     <<< SOUND 4 >>>
20 REM *  explosion and melody
30 WAIT=50
40 FOR T=1 TO 20
50 SOUND 0,INT(RND(1)*256),10,10
60 GOSUB 140:NEXT T
70 SOUND 0,0,0,0
80 WAIT=3
90 FOR I=1 TO 255 STEP 4
```

```
100 SOUND 0,I,8,10
110 GOSUB 140
120 NEXT I
130 END
140 FOR W=1 TO WAIT:NEXT W:RETURN
```

שדה התעופה

```
10 REM *   <<< AIRPORT >>>
20 REM * Original by Bill Carris
30 REM *  Modified by Ted Kahn
40 REM *     and Len Lindsay
50 DIM TEXT$(40)
60 GRAPHICS 2+16
70 SETCOLOR 0,8,RND(1)*256
80 POSITION 0,6:TEXT$="THE AIRPORT AT NI
GHT"
90 FOR LOOP=1 TO LEN(TEXT$)
100 PRINT #6;TEXT$(LOOP,LOOP);
110 FOR PAUSE=1 TO 10:NEXT PAUSE
120 NEXT LOOP
130 FOR X=227 TO 1 STEP -1
140 SOUND 0,X,8,12
150 FOR PAUSE=1 TO 16:NEXT PAUSE
160 NEXT X
170 FOR X=1 TO 200
180 SOUND 0,X,8,12
190 NEXT X
200 FOR PAUSE=1 TO 100:NEXT PAUSE
210 SOUND 0,0,0,0
220 RUN
```



קול, עיוות ועוצמה

הנה נעימה פשוטה בארבעה קולות:

```
5 REM A SIMPLE QUARTET
10 FOR X=1 TO 50
20 SOUND 0, X, 10, 8
30 SOUND 1, X+2, 10, 8
40 SOUND 2, 100-X, 10, 8
50 SOUND 3, 2*X, 10, 8
60 FOR W=1 TO 50:NEXT W
70 NEXT X
```

וזו הנעימה לאחר שקשטנו אותה בנקודות קופצות, ואחריהן תכנית פשוטה המדגימה כיצד עוותנו את הדו האמצעי ע"י שנוי המשתנה Z.

```
5 REM A SIMPLE QUARTET
10 FOR X=1 TO 50
15 GRAPHICS 3.COLOR 1
16 PLOT RND(1)*5,RND(1)*5
17 PLOT RND(1)*5+2,RND(1)*5+2
18 PLOT RND(1)*30,RND(1)*19
19 PLOT RND(1)*25,RND(1)*19
20 SOUND 0, X, 10, 8
30 SOUND 1, X+2, 10, 8
40 SOUND 2, 100-X, 10, 8
50 SOUND 3, 2*X, 10, 8
60 FOR W=1 TO 50:NEXT W
70 NEXT X
10 REM EXPERIMENT WITH DISTORTION
20 FOR X=0 TO 14 STEP 2
30 SOUND 0, 121, X, 8
40 FOR Y=1 TO 1000:NEXT Y
50 NEXT X
```

תכנית זו מדגימה את אפשרויות השנוי בעצמת הצליל. גם היא משתמשת בדו האמצעי.

```
10 REM CHANGES IN VOLUME
20 FOR W=1 TO 15
30 SOUND 0, 121, 10, W
40 FOR M=1 TO 200:NEXT M
50 NEXT W
```

על פי העקרונות של הדוגמאות הפשוטות הללו ועל הבסיס של ידיעותיך בגרפיקה, יהיה ביכלתך לבנות תכניות משלך. והנה עוד כמה נסיונות קוליים :

האטארי שלך הופך כעת, באמצעות התכנית הזו, ללוח מקשים מוסיקלי :

```
10 REM *  KEYBOARD ORGAN  **********
20 REM * SIMULATED TOY ORGAN SOUNDS*
30 REM * (TO GET LOWER TONES, PRESS*
40 REM * [CAPS-LOWR] OR THE ATARI  *
50 REM * "INVERSE VIDEO" KEY)      *
60 REM * (C) TED M. KAHN, 1980     *
70 REM ****************************
80 OPEN #1,4,0,"K"
90 PRINT "[CLEAR]PRESS ANY KEY ON THE KE
YBOARD TO GET":PRINT "A NEW TONE."
100 GET #1,N
110 FOR I=1 TO 16 STEP 0.5
120 SOUND 1,N,10,I
130 NEXT I
140 GOTO 100
```

וכעת הוא יכול להשמע כפסנתר צעצוע :



```
10 REM *  KEYBOARD PIANO  **********
20 REM * SIMULATED PIANO SOUNDS    *
30 REM * (JUST PRESS ANY KEY)      *
40 REM * (C) TED M. KAHN, 1980     *
50 REM ****************************
60 OPEN #1,4,0,"K"
70 PRINT "[CLEAR]PRESS ANY KEY ON THE KE
YBOARD TO GET":PRINT "TOY PIANO SOUNDS."
80 GET #1,N
```

```
90 FOR I=16 TO 0 STEP -1
100 SOUND 1,N,10,I
110 NEXT I
120 GOTO 80
```

וזוהי תכנית שתתן לך אקורדים :

```
10 REM * KEYBOARD CHORDS *********
20 REM * TO GET CHORDS, PRESS THE  *
30 REM * 'START','SELECT' OR       *
40 REM * 'OPTION' KEYS.            *
50 REM * (C) LEN LINDSAY, 1980     *
60 REM ***************************
70 PRINT "[CLEAR]PRESS THE 'START', 'SEL
ECT' OR":PRINT "'OPTION' KEYS FOR CHORDS
"
80 FOR Z=1 TO 3:SOUND Z,0,10,0:NEXT Z
90 N=PEEK(53279):IF N=7 THEN 90
100 RESTORE 200+N
110 FOR Z=1 TO 3:READ A:SOUND Z,A,10,6
120 NEXT Z
130 GOTO 90
203 DATA 162,108,128
205 DATA 182,121,144
206 DATA 243,96,162
```

בונה אקורדים

תכנית זו, שנכתבה ע"י טד קאהן, מאפשרת לך לבנות אקורדים משלך ולנסות הרמוניות. לשם כך דרושה ידית בקרה (joystick), שאותה צריך לחבר לשקע הראשון בצד השמאלי של המחשב. על המסך תופענה ארבע שורות, ותוך שמוש בידית הבקרה כדי להזיז את הנתב, אפשר לשלוט בגובה הצליל. כשמחליטים על הצליל הרצוי, לוחצים על הכפתור האדום וזה קובע ושומר את הצליל הזה על אחת השורות. ואז נעים לשורה אחרת לבחור צליל אחר. אפשר לבנות כך אקורדים בני שנים, שלשה או ארבעה צלילים. זוהי גם תכנית מצויינת לאלה הלומדים תיאוריה מוסיקלית והרמוניות.

```
10 REM *  <<< CHORD BUILDER >>>
20 REM * (C) 1980 By Ted M. Kahn
30 POKE 82,1
40 POKE 752,1:REM CURSOR OFF
50 DIM H$(40),SCALE$(40),PITCH(37)
60 DIM BLANKLINE$(37),LAST(4)
70 REM INITIALIZE CONSTANTS
80 H$="1[W][S][W][S][S][W][S][W][S][W][S
]2[W][S][W][S][S][W][S][W][S][W][S]3[W][
S][W][S][S][W][S][W][S][W][S]4"
90 FOR I=1 TO 4:LAST(I)=1:NEXT I
100 SCALE$="C D EF G A BC D EF G A BC D
EF G A BC "
110 BLANKLINE$="
               "
120 RESTORE 340
130 FOR I=1 TO 37
140 READ P:PITCH(I)=P
150 NEXT I
160 GOSUB 570:REM INSTRUCTIONS
170 FOR Y=1 TO 4
180 GOSUB 380
190 NEXT Y
200 REM POSITION JOYSTICK
210 FOR VOICE=0 TO 3 STEP 1
220 MOVE=0:X=LAST(VOICE+1)
230 POSITION X,20-(VOICE*5)
240 SOUND VOICE,PITCH(LAST(VOICE+1)),10,
6
250 IF STRIG(0)=0 THEN 310
260 POSITION X,20-(VOICE*5):PRINT "^";:G
OSUB 500:PRINT "[LEFT] ";:GOSUB 500
270 IF STICK(0)=7 AND X<37 THEN X=X+1:PR
INT "[RIGHT]":GOSUB 440:MOVE=MOVE+1:IF M
OVE=1 THEN GOSUB 520
280 POSITION X,20-(VOICE*5):PRINT "^";:G
OSUB 500:PRINT "[LEFT] ";:GOSUB 500
290 IF STICK(0)=11 AND X>1 THEN X=X-1:PR
INT "[LEFT]":GOSUB 440:MOVE=MOVE+1:IF MO
VE=1 THEN GOSUB 520
300 GOTO 250
310 POSITION X,20-(VOICE*5):PRINT "[T]";
:GOSUB 470:LAST(VOICE+1)=X
320 NEXT VOICE
```

```
330 POP :GOTO 200
340 DATA 243,230,217,204,193,182,172,162
,153,144,136,128
350 DATA 121,114,108,102,96,91,85,81,76,
72,68,64
360 DATA 60,57,53,50,47,45,42,40,37,35,3
3,31
370 DATA 29
380 REM DRAW SCALE LINE
390 POSITION 1,(Y*5)-3
400 PRINT "VOICE ";CHR$(53-Y+128)
410 PRINT SCALE$
420 POSITION 1,(Y*5)-1:PRINT H$
430 RETURN
440 REM SOUND & MOVE POINTER
450 SOUND VOICE,PITCH(X),10,6
460 RETURN
470 REM WAIT FOR SYCHRONIZATION
480 FOR WW=1 TO 300:NEXT WW
490 RETURN
500 FOR W1=1 TO 5:NEXT W1
510 RETURN
520 REM ERASE PREVIOUS NOTE
530 POSITION 1,20-(VOICE*5)
540 PRINT BLANKLINE$;
550 POSITION X,20-(VOICE*5)
560 RETURN
570 REM INSTRUCTIONS FOR USE
580 PRINT "[CLEAR]Use a joystick in port
 #1 to change":PRINT "notes.  Press trig
ger to move to next"
590 PRINT "voice.":PRINT :PRINT "PRESS S
TART WHEN READY"
600 IF PEEK(53279)<>6 THEN 600
610 PRINT "[CLEAR]":POSITION 14,0
620 PRINT "CHORD BUILDER"
630 RETURN
```

צירופי צבע וצליל

הנה עוד הדגמה של ערבוב אפקטים פשוטים של צבע וצליל. וניתן, קרוב לודאי, לשנות את הגרפיקה כך שהאפקטים יהיו יותר מענינים !

```
0 REM *   <<< COLOR & SOUND >>>
5 REM * From collection at ATARI
10 N1=0:N2=0:GOSUB 1000
20 GOSUB 2000
30 GOSUB 3000
40 GOSUB 1000
50 GOSUB 2000
60 N2=61:GOSUB 3000
70 N2=46:GOSUB 1000
80 GOSUB 2000
90 GOSUB 3000
100 N1=48:N2=41:GOSUB 1000
110 N1=46:N2=36:GOSUB 2000
120 N1=41:N2=34:GOSUB 3000
130 N1=36:N2=31:GOSUB 1000
140 GOSUB 2000
150 GOSUB 3000
160 N1=48:N2=41:GOSUB 1000
170 GOSUB 2000
180 GOSUB 3000
190 N1=46:N2=36:GOSUB 1000
200 GOSUB 2000
210 N1=48:GOSUB 3000
220 N1=54:N2=34:GOSUB 1000
230 N1=61:N2=36:GOSUB 2000
240 N1=69:N2=41:GOSUB 3000
250 N1=73:N2=46:GOSUB 1000
260 GOSUB 2000
270 N1=69:N2=54:GOSUB 3000
280 SOUND 0,183,10,8
290 SOUND 1,82,10,8
300 SOUND 2,69,10,8
310 SOUND 3,61,10,8
320 FOR W=1 TO 50:NEXT W
330 N1=61:N2=0:N3=0:GOSUB 3000
340 N1=73:N2=46:GOSUB 1000
```

 זוהי אחת מאוסף תכניות מתוך Graphics/Sound Demonstration disk, (APX-20028) שניתן להשיגה מ Atari Program Exchange
P.O.Box 427, Sunnyvale, Ca.94082
(מידע מענין על אפקטים של צבע וצליל ישנו במדריך לדיסק הזה). התכנית הזו מודפסת כאן ברשות מאטארי בע"מ

```
350 GOSUB 2000
360 N1=46:N2=0:GOSUB 3000
370 N1=61:N2=41:GOSUB 1000
380 N1=46:N2=36:GOSUB 2000
390 N1=41:N2=34:GOSUB 3000
400 N1=36:N2=31:GOSUB 1000
410 GOSUB 2000
420 N1=46:GOSUB 3000
430 N1=48:N2=41:GOSUB 1000
440 GOSUB 2000
450 N1=0:N2=36:GOSUB 3000
460 N1=54:N2=34:GOSUB 1000
470 GOSUB 2000
480 N1=0:GOSUB 3000
490 N1=61:N2=34:GOSUB 1000
500 N2=36:GOSUB 2000
510 N1=69:N2=41:GOSUB 3000
520 N1=73:N2=61:GOSUB 1000
530 N2=46:GOSUB 2000
540 N1=82:N2=61:GOSUB 3000
550 SOUND 0,183,10,8
560 SOUND 1,92,10,8
570 SOUND 2,46,10,8
580 FOR W=1 TO 75:NEXT W
590 RUN
1000 SOUND 0,183,10,8
1010 SOUND 1,N1,10,8
1020 GRAPHICS 3+16:COLOR 1
1030 SETCOLOR 0,8,2
1040 PLOT 0,12:DRAWTO 5,12
1050 SOUND 2,N2,10,8
1060 FOR W=1 TO 25:NEXT W:RETURN
2000 SOUND 0,137,10,8
2010 SOUND 1,N1,10,8
2020 GRAPHICS 3+16:COLOR 1
2030 SETCOLOR 0,1,8
2040 PLOT 0,12:DRAWTO 10,12
2050 SOUND 2,N2,10,8
2060 FOR W=1 TO 25:NEXT W:RETURN
3000 SOUND 0,145,10,8
3010 SOUND 1,N1,10,8
3020 GRAPHICS 3+16:COLOR 1
3030 SETCOLOR 0,4,2
3040 PLOT 0,12:DRAWTO 15,12
3050 SOUND 2,N2,10,8
3060 FOR W=1 TO 25:NEXT W:RETURN
```

ג'ז- שילוב צבע וצליל

התכנית הזו, שנכתבה ע"י מתכנתים באטארי, מראה כיצד אפשר לשלב שתי תפיסות. היא משתמשת באותן צורות מלבניות שבתכנית בשם ALBERS שנציג בפרק ח', אבל מוסיפה אקורד של מוסיקת בלוז ואת האפשרות לשנות את המהירות (טמפו). ככל שהמספר קטן יותר (בין 1 ל-9), כך גדלה מהירות הנגינה.

```
10 REM *       <<< JAZZ >>>
20 REM * by Software Group at ATARI
30 OPEN #3,4,0,"K:":GRAPHICS 17
40 CHORD=1:THNOT=1:PTR=1
50 DIM COLPTR(4)
60 ? #6:? #6;"      ENTER TEMPO ";
70 IF PEEK(764)=255 THEN 70
80 GET #3,T:PRINT #6;T-48:GET #3,O:CLOSE
 #3:TEMPO=T-48
90 DIM BASE(3,4)
100 DIM LOW(3)
110 DIM LINE(16)
120 DIM JAM(3,7)
130 DIM COLS(7)
140 DIM LUMS(7)
150 FOR X=1 TO 3
160 FOR Y=1 TO 4
170 READ A:BASE(X,Y)=A
180 NEXT Y
190 NEXT X
200 FOR X=1 TO 3:READ A:LOW(X)=A
210 NEXT X
220 FOR X=1 TO 16:READ A:LINE(X)=A:NEXT
X
230 FOR X=1 TO 3
240 FOR Y=1 TO 7
250 READ A:JAM(X,Y)=A:NEXT Y:NEXT X
260 FOR X=1 TO 7:READ A:COLS(X)=A:NEXT X
270 FOR X=1 TO 7:READ A:LUMS(X)=A:NEXT X
280 FOR X=1 TO 4:COLPTR(X)=1:NEXT X
290 GOSUB 810
300 GOSUB 470
310 T=T+1
320 GOSUB 350
330 GOSUB 750
```

△ GRAPHICS 17 בשורה 30 מקביל ל-GRAPHICS 1+16 (כלומר, מסך מלא)

```
340 GOTO 300
350 REM PROCESS HIGH STUFF
360 IF RND(0)<0.25 THEN RETURN
370 IF RND(0)<0.5 THEN 420
380 NT=NT+1
390 IF NT>7 THEN NT=7
400 GOSUB 980
410 GOTO 450
420 NT=NT-1
430 IF NT<1 THEN NT=1
440 GOSUB 1040
450 SOUND 2,JAM(CHORD,NT),10,NT*2
460 RETURN
470 REM PROCESS BASE STUFF
480 IF BASS=1 THEN 550
490 BDUR=BDUR+1
500 IF BDUR<>TEMPO THEN 520
510 BASS=1:BDUR=0
520 SOUND 0,LOW(CHORD),10,4
530 SOUND 1,BASE(CHORD,THNOT),10,4
540 RETURN
550 SOUND 0,0,0,0
560 SOUND 1,0,0,0
570 BDUR=BDUR+1
580 IF BDUR<>1 THEN 660
590 BDUR=0:BASS=0
600 THNOT=THNOT+1
610 IF THNOT<>5 THEN 660
620 THNOT=1
630 PTR=PTR+1
640 IF PTR=17 THEN PTR=1
650 CHORD=LINE(PTR)
660 RETURN
670 DATA 162,144,136,144,121,108,102,108
,108,96,91,96
680 DATA 243,182,162
690 DATA 1,1,1,1,2,2,2,2,1,1,1,1,3,2,1,1
700 DATA 60,50,47,42,40,33,29
710 DATA 60,50,45,42,40,33,29
720 DATA 81,68,64,57,53,45,40
730 DATA 0,2,4,8,12,15,0
740 DATA 0,2,2,2,2,2,12
750 REM SET COLOR POINTERS TO COLORS
760 FOR X=1 TO 3
```

```
770 SETCOLOR X-1,COLS(COLPTR(X)),LUMS(CO
LPTR(X))
780 NEXT X
790 SETCOLOR 4,COLS(COLPTR(4)),LUMS(COLP
TR(4))
800 RETURN
810 GRAPHICS 7+16
820 COLOR 2
830 FOR X=20 TO 140
840 PLOT X,20
850 DRAWTO X,70
860 NEXT X
870 COLOR 1
880 FOR X=40 TO 120
890 PLOT X,30
900 DRAWTO X,60
910 NEXT X
920 COLOR 0
930 FOR X=60 TO 100
940 PLOT X,40
950 DRAWTO X,50
960 NEXT X
970 RETURN
980 REM SHIFT COLORS OUT
990 FOR X=4 TO 2 STEP -1
1000 COLPTR(X)=COLPTR(X-1)
1010 NEXT X
1020 COLPTR(1)=NT
1030 RETURN
1040 FOR X=1 TO 3
1050 COLPTR(X)=COLPTR(X+1)
1060 NEXT X
1070 COLPTR(4)=NT
1080 RETURN
```

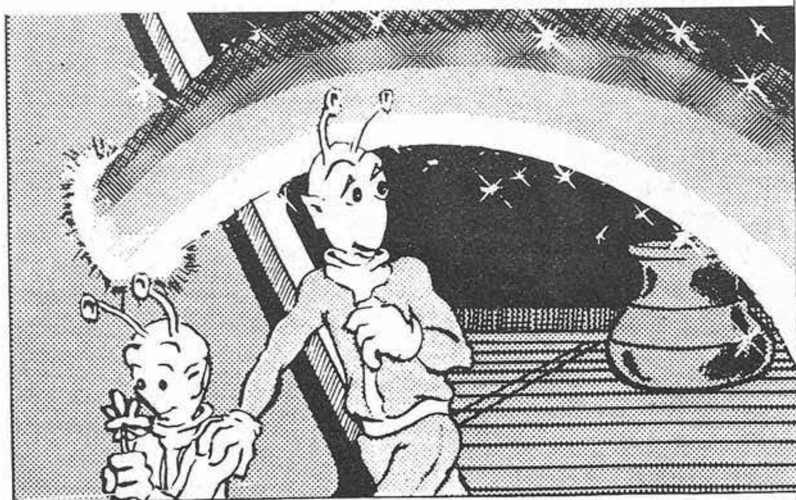
נוראות המלחמה (מראות וקולות)

תכנית הקול האחרונה כאן היא אוסף של אפקטים חזותיים וקוליים שאפשר לשלבם במשחקים או פעילות בידורית אחרת. היא גם מראה שדמויות מצוירות ובעות אינן דוקא דבר מסובך - הרמיזה היא שקובעת. זוהי גם דוגמה לשילוב של מימדי צבע, צליל, הנפשה - מה שרק תרצו !

```
10 REM * <<< THE HORRORS OF WAR >>>
20 REM *  Sound and visual effects
30 REM *         from ATARI
40 DIM A$(10)
50 POKE 752,0
60 R=INT(RND(0)*4)+1
70 ON R GOTO 90,250,400,560
80 GOTO 60
90 REM *  BOMB  *
100 GRAPHICS 3+16
110 FOR N=20 TO 125
120 SOUND 0,N,10,10:X=INT(N/6)
130 COLOR 0:PLOT 10,X-1
140 COLOR 1:PLOT 10,X
150 FOR Z=1 TO 15:NEXT Z
160 NEXT N
170 COLOR 0:PLOT 10,X
180 FOR N=0 TO 15
190 SETCOLOR 4,4,15-N
200 SOUND 0,20,0,15-N
210 SOUND 1,227,6,15-N
220 FOR Z=1 TO 30:NEXT Z
230 NEXT N
240 GOTO 80
250 REM *  HELICOPTER  *
260 GRAPHICS 0
270 POKE 752,1
280 A$="HELICOPTER"
290 FOR N=1 TO 25
300 POSITION 28-N,10:PRINT #6;A$
310 SOUND 0,10,0,12
320 FOR Z=1 TO 5:NEXT Z
330 SOUND 0,0,0,0
340 FOR Z=1 TO 20:NEXT Z
350 SOUND 1,220,6,8
360 POSITION 28-N,10:PRINT #6;"
 "
370 NEXT N
380 SOUND 1,0,0,0
390 GOTO 80
400 REM *  MACHINE GUN  *
410 GRAPHICS 7+16
420 FOR N=1 TO 40
430 FOR Z=0 TO 15 STEP 3
```

```
440 SOUND 0,60,6,15-Z
450 NEXT Z
460 SOUND 0,0,0,0
470 FOR Z=1 TO 5:NEXT Z
480 Y=INT(RND(0)*6)
490 IF Y1>85 OR Y1<5 THEN Y1=40
500 IF Y>3 THEN Y1=Y1+Y
510 IF Y<=3 THEN Y1=Y1-Y
520 COLOR 1:PLOT 4*N-1,Y1
530 NEXT N
540 Y1=40
550 GOTO 80
560 REM *  EXPLOSIONS  *
570 GRAPHICS 7+16
580 FOR N=0 TO 15
590 SETCOLOR 4,6,15-N
600 SOUND 0,100,0,15-N
610 FOR Z=1 TO 15:NEXT Z
620 NEXT N
630 FOR Z=1 TO 20:NEXT Z
640 FOR N=0 TO 15
650 SETCOLOR 4,10,6
660 SOUND 0,80,0,15-N
670 FOR Z=1 TO 5:NEXT Z
680 NEXT N
690 FOR N=1 TO 15
700 SETCOLOR 4,2,15-N
710 SOUND 0,150,0,15-N
720 FOR Z=1 TO 20:NEXT Z
730 SETCOLOR 4,8,15-N
740 NEXT N
750 GOTO 80
```

פרק ח'



צבע

השליטה בשמוש בצבע באטארי 400 ואטארי 800 היא קצת יותר קשה מהשליטה בצליל ובגרפיקה. אנו נציג בפניך כמה דרכים לשימוש ושליטה בצבע שניתן לשלבן בתכניותיך, אבל יחד עם זאת עליך להוועץ בספר

ATARI 400/800 BASIC Reference Manual

ובמקורות אחרים אם ברצונך להתעמק יותר בתכנות בצבע.

אנו מציעים לתקתק את התכנית הבאה ולשמור אותה על סרט או תקליטון לפני שמתחילים בנסויים אישיים בתכניות בצבע. התכנית הזו מציגה 128 צבעים שמחשב אטארי יכול להפיק. לכל אחד מבין 16 הצבעים (הממוספרים 0-15) יש 8 דרגות או עצמות של בהירות. ע"י המקשים המיוחדים (SELECT, OPTION -ו START) אפשר לראות כיצד מופיעות אותיות גדולות בצבעים השונים הללו, וכיצד נראה צבע הרקע בכל אחד מן הגוונים (עם SETCOLOR 4 בדוגמה הזאת). אחרי שהשתמשת בתכנית כמה פעמים, אפשר לבטל את השורות 200-330. צריך לבחור ב GRAPHICS 1 או 2, ולהחליף את ה-M בשורה 60 במצב הגרפי המסוים שבחרת.

```
10 REM *    <<< COLORLET >>>
20 REM * Demo of 128 ATARI colors
30 REM * using large text & background
40 REM (C)1980 LEN LINDSAY & TED KAHN
50 GOSUB 200
60 GRAPHICS M
70 PRINT #6;"CAPS"
80 PRINT #6;"REVERSE CAPS"
90 PRINT #6;"lower case"
100 PRINT #6;"reverse lower "
110 FOR XR=0 TO 4
120 PRINT "[CLEAR]THIS IS SETCOLOR ";XR;
"..."
130 PRINT "HIT OPTION TO CHANGE COLOR LU
MINANCE"
140 PRINT "HIT SELECT TO CHANGE COLOR"
150 PRINT "HIT START TO DO NEXT COLOR RE
GISTER";
160 GOSUB 340
170 FOR Z=1 TO 99:NEXT Z
180 NEXT XR
```

```
190 GOTO 60
200 REM ** CHOOSE LETTER SIZE **
210 PRINT "[CLEAR]THIS IS A DEMONSTRATIO
N OF"
220 PRINT "          ATARI COLORS"
230 PRINT "USING A LARGE OR EXTRA LARGE
ALPHABET"
240 PRINT :PRINT "IT SHOWS HOW  TEXT TYP
ED IN (GR.0) AS":? "       * CAPITAL LETT
ERS"
250 PRINT "       * REVERSE CAPITAL LETTE
RS"
260 PRINT "       * lower case letters"
270 PRINT " and   * reverse lower case le
tters"
280 PRINT :PRINT "ARE ALL CAPITAL LETTER
S AND":? "CHANGE COLORS WHEN DISPLAYED I
N"
290 PRINT "GRAPHICS MODE 1 OR 2.":PRINT
:? "(IT ALSO SHOWS HOW THE BACKGROUND CA
N CHANGE COLOR)"
300 PRINT :PRINT :? "CHOOSE A GRAPHICS M
ODE FOR TEXT SIZE":? "(1=LARGE, 2=EXTRA
LARGE)";
310 TRAP 300
320 INPUT M:IF M<1 OR M>2 THEN 300
330 RETURN
340 A=PEEK(53279):IF A=7 THEN 340:REM NO
NE HIT YET
350 IF A=6 THEN 430:REM **NEW COLOR REGI
STER (RETURN)**
360 IF A=5 THEN XC=XC+1:IF XC>15 THEN XC
=0:REM ** NEW COLOR **
370 IF A=3 THEN XL=XL+2:IF XL>14 THEN XL
=0:REM ** NEW LUMINANCE LEVEL **
380 SETCOLOR XR,XC,XL
390 POSITION 1,8:PRINT #6;"setcolor
    ";
400 POSITION 10,8:PRINT #6;XR;",";XC;","
;XL
410 IF PEEK(53279)<>7 THEN 410
420 GOTO 340
430 POSITION 10,8:PRINT #6;"         ":RE
TURN
```

יש שתי פקודות צבע ב BASIC של אטארי :

COLOR X
SETCOLOR Y, Z, W

הפקודות האלה מבצעות דברים שונים, תלוי באיזה מצב גרפי נמצאים. כשמדליקים את המחשב נמצאים במצב גרפי 0, וזהו המצב הגרפי הרגיל לתכנות. למעשה, חלון הטקסט נשאר במצב 0 גם במצבים הגרפיים 1 עד 8, למרות שבשאר המסך אפשר לצייר או ליצור אותיות גדולות.

במצב הגרפי 0 האותיות בצבע לבן, הרקע כחול וגבולות המסך (המסגרת) הם שחורים. ע"י הפקודה SETCOLOR Y,Z,W אפשר לשנות את צבע האותיות, הרקע והגבולות. △

בפקודה SETCOLOR Y,Z,W, Y מורה למחשב לשנות את הצבע של האותיות, של הרקע או של הגבולות. אם הערך של Y הוא 2, ישתנה צבע הרקע. אם Y =4 , נשנה את צבע המסגרת. Z , המשתנה השני, מציין באיזה צבע רוצים להשתמש.
ערכו של Z נע בין 0 ל-15 עם מספרי צבע קבועים, עפ"י הטבלה הבאה : △△

0	-	אפור
1	-	כתום בהיר (זהב)
2	-	כתום
3	-	כתום-אדום
4	-	ורוד
5	-	חום-אדום
6	-	כחול-סגול
7	-	כחול
8	-	כחול
9	-	כחול בהיר
10	-	טורקיז
11	-	כחול-ירוק
12	-	ירוק
13	-	ירוק-צהוב
14	-	ירוק-כתום
15	-	כתום בהיר



△ חשוב לזכור : הפקודה SETCOLOR Y,Z,W פועלת באופן שונה במצבים הגרפיים השונים.

△△ גוני הצבע ישתנו מעט עם כל מכשיר טלביזיה או מסך מחשב

השלישי מבין המשתנים של SETCOLOR Y,Z,W, W; קובע את הבהירות במסך. הוא נע בתחום המספרים הזוגיים מ-0 עד 14 (כלומר 14, 12, 10, 8, 0,2,4,6).
ככל שהמספר גבוה יותר, כך יהיה הרקע בהיר יותר.
כאשר W=0, תהיינה האותיות בהירות מאד והמסך יראה עכור.
כאשר W=14, יהיו האותיות והמסך בהירים מאד, והאותיות גם הן תראינה בצבע קרוב לזה של המסך.
כאשר W=10, האותיות תהיינה באותו צבע ובאותה דרגת בהירות של המסך כך שהן תתמזגנה עם המסך וכמעט שאי אפשר לראותן.
הנה כמה תכניות פשוטות שמדגימות את היבטי הצבע ב- GRAPHICS 0 :

1. שנוי צבעי הרקע באופן אקראי עם אותיות בהירות, תוך שמוש בפקודה

```
SETCOLOR 2, INT(RND (1)*16), 0
```

```
5 REM BRIGHT WHITE LETTERS WITH COLOR CHANGES
10 SETCOLOR 2,INT(RND(1)*16), 0
20 PRINT "HELP ME I'M LOST ON A MAP";
30 FOR X=1 TO 200:NEXT X
40 GOTO 10
```

2. שנוי צבע הרקע באופן אקראי עם אותיות בגוונים-

```
SETCOLOR 2, INT(RND(1)*16), 14
```

```
5 REM BRIGHT LETTERS THE COLOR OF BACKGROUND
10 SETCOLOR 2,INT(RND(1)*16), 14
20 PRINT "HELP ME I'M LOST ON A MAP";
30 FOR X=1 TO 200:NEXT X
40 GOTO 10
```

3. אותיות בלתי נראות, שנויי צבע אקראיים -

```
SETCOLOR 2, INT(RND(1)*16), 10
```

```
5 REM LETTERS AND BACKGROUND SAME LUMINANCE
6 REM SO LETTERS CANNOT BE SEEN
10 SETCOLOR 2, INT(RND(1)*16), 10
20 PRINT "HELP ME I'M LOST ON A MAP ";
30 FOR X=1 TO 200:NEXT X
40 GOTO 10
```

4. שנוי צבעי המסגרת -

```
SETCOLOR 4, INT(RND(1)*16), 14
```

```
5 REM NOW THE BORDER CHANGES
10 SETCOLOR 4, INT(RND(1)*16), 14
20 PRINT "HELP ME I'M LOST ON A MAP ";
30 FOR X=1 TO 200:NEXT X
40 GOTO 10
```

וכעת גרסה צבעונית של המשחק הראשון שהצגנו בספר : ניחוש מספר. לתכנית זו, המשתמשת במה שידוע לך עכשו על הצבע באטארי, נוכל לקרוא :

מלבוש צבעוני לניחוש מספרי

```
5 REM COLOR DRESSING FOR NUMBER GUESSING
7 LET Y=INT(RND(1)*10)+1
8 FOR X=1 TO 20
10 PRINT "GUESS A NUMBER FROM 1 TO 10"
20 SETCOLOR 2,RND(1)*16, 14
30 SETCOLOR 4,RND(1)*16, 14
40 NEXT X
50 PRINT "[CLEAR]"
55 PRINT "YOUR GUESS IS ?"
60 INPUT A
70 IF A=Y THEN GOTO 1000
80 GOTO 8
1000 FOR W=1 TO 20
1010 SETCOLOR 2,RND(1)*16, 14
1020 SETCOLOR 4,RND(1)*16, 14
1030 PRINT "YOU GOT IT";
1040 NEXT W
1050 PRINT "[CLEAR]"
1055 FOR T=1 TO 500:NEXT T
1060 GOTO 7
```



בשלב זה נראה לנו שיש ביכלתך לשלב את פקודת SETCOLOR Y,Z,W במשחקים שלך במצב הגרפי 0. עבור המצבים הגרפיים 3 עד 8 עליך להשתמש ב- SETCOLOR 4 ,Z,W כדי לשנות את הרקע, ובפקודה COLOR X כדי לשנות את צבע הנקודות שתתקבלנה ע"י פקודת PLOT. X יכול להיות 1, 2 או 3

נקודות צהובות COLOR 1
נקודות בירוק בהיר COLOR 2
נקודות כחולות COLOR 3

זוהי תכנית פשוטה שמפיקה את כל שלשת הצבעים בנקודות אקראיות על המסך והתוצאה היא מספר אינסופי של צורות :

מוזאיקה צבעונית ואקראית

```
1 REM *** A RANDOM COLOR MOSAIC ***
2 REM LEN LINDSAY 12/79
5 GRAPHICS 3+16
10 FOR X=1 TO 3
20 COLOR X
30 PLOT INT(RND(0)*37),INT(RND(0)*23)
40 NEXT X
10 GOTO 10
```

אם נוסיף פקודת SETCOLOR 4,Z,W בשורה 6, נוכל לשנות את צבע הרקע ואת צבע הנקודות כאחד. למשל, SETCOLOR 4,5,0 בשורה 6 יתן מוזאיקה כחולה,לבנה וצהובה על רקע חום-אדום. אפשר להשתמש ב COLOR X וב - SETCOLOR Y,Z,W כדי ליצור צורות בארבעה צבעים על המסך, וכדי לקשט את התכניות.
המצבים הגרפיים 4, 5, 6 ו-7 פועלים בדיוק באותו האופן כמו GRAPHICS 3.

ואם תרצו לראות את אפשרויות הצבע האחרות של האטארי הנה תכנית הדגמה של 128 צרופי הצבע/הבהירות, שאנו מכנים בשם הקשת של אטארי :

```
10 REM <<< RAINBOW >>>
20 REM * by the Software Engineers
30 REM * at ATARI INC.
40 PRINT "[CLEAR]THIS IS A PROGRAM WHICH
 DISPLAYS":? "ALL 128 OF YOUR ATARI'S CO
LORS ":? "ON THE SCREEN AT THE SAME TIME
!"
50 PRINT :PRINT "(IT DOESN'T USE NORMAL
GRAPHICS MODES)"
60 PRINT :PRINT "THERE ARE 16 COLOR BARS
, EACH WITH":? "8 DIFFERENT SHADES OR LU
MINANCES"
```

```
70 ? :? "IT USES VERY ADVANCED FEATURES
OF THE":? "ATARI'S DISPLAY SYSTEM..."
80 FOR WAIT=1 TO 1800:NEXT WAIT
90 DIM C$(24)
100 SETCOLOR 2,0,0:POKE 752,1:? "[CLEAR]
"
110 FOR I=1 TO 24
120 READ D
130 C$(I,I)=CHR$(D)
140 NEXT I
150 D=USR(ADR(C$))
160 END
170 DATA 162,0,173,11,212,201,32,208,249
,141
180 DATA 10,212,142,24,208,232,232,208,2
46,142
190 DATA 24,208,240,232
```

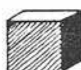
כפי שודאי כבר הבחנת, השתמשנו בצבע בתכניות לאורך כל הספר. אולי כדאי לחזור ולראות כיצד שלבנו את הצבע בתוך התכניות, עכשו כשיש לך כבר הבנה מסוימת של הפקודות COLOR X ו- SETCOLOR Y,Z,W.

הבה נתבונן בכמה תכניות אחרות המדגימות את אפשרויות הצבע של המחשב האישי שלך.

קוביה צבעונית בשלושה מימדים

התכנית הבאה, שנכתבה בידי דייב ת'ורנברג, מראה כיצד ניתן להשתמש בדרגות הבהירות השונות של האטארי כדי לתת אפקטים גרפיים בשלושה מימדים. בתכנית זו, דייב "מצליל" (מלשון צל) את הפאות השונות של קוביה. אחרי טעינת התכנית, עליך פשוט ללחוץ על מקש כלשהו בלוח המקשים, ולהמשיך ללחוץ על כל מיני מקשים כדי לראות מה קורה ! (לתשומת לבך: GRAPHICS 23 בשורה 50 הוא בעצם GRAPHICS 7+16, כלומר, GRAPHICS 7 ללא חלון טקסט.

התכנית הודפסה ברשות
COMPUTE ! The Journal for Progressive Computing
שהוא מקור מצויין לתכניות ורעיונות למשתמשי אטארי.
COMPUTE ! יוצא לאור ע"י
Small Systems Services, Inc., P.O.Box 5406
Greensboro, N.C. 27043



```
10 REM <<< 3-D COLOR CUBE >>>
20 REM * By Dave Thornburg
30 PRINT "[CLEAR]PRESS DIFFERENT KEYS TO
 ":PRINT "CHANGE COLORS."
40 FOR WW=1 TO 1200:NEXT WW
50 GRAPHICS 23
60 OPEN #1,4,0,"K:"
70 FOR I=0 TO 4:SETCOLOR I,9,4:NEXT I
80 X0=48:Y0=36
90 COLOR 1
100 FOR I=0 TO 40
110 PLOT X0,Y0+I:DRAWTO X0+40,Y0+I
120 NEXT I
130 COLOR 2
140 FOR I=1 TO 24
150 PLOT X0+I,Y0-I:DRAWTO X0+I+40,Y0-I
160 NEXT I
170 COLOR 3
180 FOR I=1 TO 24
190 PLOT X0+40+I,Y0-I:DRAWTO X0+40+I,Y0+
40-I
200 NEXT I
210 FOR I=0 TO 2
220 GET #1,A
230 IF A<48 THEN A=48
240 SETCOLOR I,1,2*(A-48)
250 NEXT I
260 POKE 77,0:GOTO 210
```

משולש- הדגמה פשוטה של שרטוט בצבע

תכנית זו היא סדרה של וריאציות על נושא - במקרה זה, המשולש. היא מראה כיצד אפשר להשתמש בפקודות פשוטות מאד ולהפיק תוצאות יפות. היא גם מראה כיצד אפשר להפיק צבעים חדשים תוך שמוש בטכניקה הקרויה "color artifact".

```
10 REM *       <<< TRIANGLE >>>
20 REM *        by ATARI staff
30 GRAPHICS 7+16
40 E=INT(300*RND(1))
50 D=INT(300*RND(1))
60 C=1
70 COLOR C
80 B=39
90 A=79
100 FOR S=1 TO D STEP E
110 FOR X=A TO B STEP -2
120 PLOT 80,A-X
130 DRAWTO 80+X,INT(A/S)
140 DRAWTO 80,X
150 DRAWTO 80-X,INT(A/S)
160 DRAWTO 80,A-X
170 COLOR C
180 NEXT X
190 C=C+1
200 NEXT S
210 SETCOLOR 0,T,2
220 T=T+1
230 GOTO 40
```

אמנות מודרנית על-פי ג'וזף אלברס

תכנית זו מדגימה כמה יחסים בין צבעים תוך שמוש בצורה שנעשתה פופולרית בידי האמן בן המאה העשרים, ג'וסף אלברס. ע"י התמקדות במוטיב מלבני מסוים, יכול היה אלברס לחקור מימדים רבים של צבע כמו עומק, למרות שציוריו נראו פשוטים למדי. ע"י שנוי גודלם של המלבנים והמרווחים ביניהם, אפשר לחקור את היחסים האלה בצורה יותר מלאה.

```
0 REM <<< ALBERS >>>
10 REM * Color paintings a la
20 REM * Josef Albers (20TH C. ART)
30 REM * By Ted M. Kahn 1980
40 SETCOLOR 0,1,2
50 SETCOLOR 1,2,4
60 SETCOLOR 2,3,8
70 X1=10:Y1=0
80 GRAPHICS 3+16
90 OSIZE=20:NSIZE=20
```

```
100 COLOR 1
110 GOSUB 290
120 NSIZE=16
130 GOSUB 360
140 COLOR 2
150 GOSUB 290
160 NSIZE=12
170 GOSUB 360
180 COLOR 3
190 GOSUB 290
200 FOR I=1 TO 15
210 FOR LUM=0 TO 14 STEP 2
220 SETCOLOR 0,I,LUM+2
230 SETCOLOR 1,I+1,LUM+4
240 SETCOLOR 2,I+2,LUM+6
250 GOSUB 410
260 NEXT LUM
270 NEXT I
280 GOTO 200
290 REM * DRAW SQUARE
300 FOR LINE=1 TO NSIZE+1
310 PLOT X1+LINE,Y1
320 DRAWTO X1+LINE,Y1+NSIZE-1
330 NEXT LINE
340 OSIZE=NSIZE
350 RETURN
360 REM * DETERMINE NEW SQUARE SIZE
370 X1=X1+(OSIZE-NSIZE)/2
380 Y1=Y1+OSIZE-NSIZE-1
390 RETURN
400 REM * PAUSE
410 FOR WW=1 TO 800:NEXT WW:RETURN
```

כיצד להשתמש בכפתור כיוון כדי לשחק עם צבע וצליל

זוהי תכנית מאד קצרה, אשר מדגימה כיצד אפשר להשתמש בכפתור הכוון כדי לשלוט בתצוגת הצבע והקול.
שורות 110 ו-140 מפקחות על האפקטים של צבע וקול ע"י הבחנה מוקדמת בשנויים בכוונים ובנקודה שבה נלחץ הכפתור האדום, בהתאמה. כפתור הכוון מתחבר לשקע הראשון במחשב.

```
10 REM *     <<< COLOR DOTS >>>
20 REM *  by Bill Carris  (ATARI)
30 REM *      REQUIRES PADDLES
40 TRAP 100:GRAPHICS 0
50 PRINT "[CLEAR]THIS PROGRAM SHOWS OFF
THE ATARI'S":? "COLORS, USING THE PADDLE
 CONTROLLERS"
60 PRINT "TO CONTROL BOTH COLOR AND DYNA
MIC DOT":? "PATTERNS.  IT ALSO HAS SOUND
 EFFECTS,"
70 PRINT "CONTROLLED FROM THE RED BUTTON
S ON ":? "THE PADDLES.":? :? "PRESS THE
START BUTTON WHEN YOU'RE "
80 PRINT "READY TO GO";
90 IF PEEK(53279)<>6 THEN 90
100 SETCOLOR 2,A,2
110 LET A=PADDLE(0)/6.2
120 IF A<38 THEN POKE 82,A:PRINT "[T][T]
"
130 SOUND 0,A,10,8
140 IF PTRIG(0)=0 THEN SOUND 0,A,6,A
150 GOTO 100
```

אפילוג להנפשה, גרפיקה וצבע

יש עדיין הרבה טריקים מתקדמים שמחכים לך, כולל
"Player-Missile graphics"
ונפלאות אחרות של הרכיב האלקטרוני החדש GTIA.
למתכנתים יותר מתקדמים, במיוחד אלה המעוניינים
בתכנות ב- 6502 Assembly language, אנו ממליצים
על הספר: De Re Atari- A Guide to Effective Programming

אפשר להשיגו אצל

Atari Program Exchange (APX),
Atari Inc., P.O. Box 427
Sunnyvale, Ca. 94086

נספח I

שגיאות והודעות שגיאה:
כיצד יעזרו לך "לדבג" את תכניתך △

כפי שהזכרנו קודם לכן, טעויות ושגיאות הן חלק רגיל ונורמלי של תהליך התכנות, כמו בכל תחום של למידה יצירתית. ככל שיתפתחו הרעיונות האישיים שלך למשחקים, כך ודאי יגיעו אל מעבר לתכניות בספר זה וללא ספק תהיינה לך כפי שהיו לנו, שגיאות. מחשב אטארי כבר תוכנן כך שיעזור לך לגלות חלק מהשגיאות הללו. הוא עושה זאת בשתי דרכים :

1) כשמתקתקים את התכנית, שורה אחרי שורה, נבדקות השורות ע"י תכנית הנקראת ATARI BASIC Interpreter. תכנית זו בודקת אם המשפטים שתקתקת הם בהירים ומובנים ב BASIC של אטארי. אם מתברר שאינם כאלה, יתן המחשב הודעת שגיאה, "ERROR" וכן יציב את הנתב על האות או התו שנראו לו כמקור הבעיה. תהליך זה פועל יפה מאד לסוגי שגיאות פשוטות, כמו שגיאות כתיב במלות מפתח של BASIC (למשל, "PINT" במקום "PRINT").

2) הסוג השני של שגיאות מתגלה רק בשלב הרצת התכנית ; לכן הן נקראות "שגיאות ריצה", בעוד שהסוג הראשון נקרא "שגיאות תחביריות" (מפני שהתחביר של המשפט ב BASIC שמופיע בתכנית אינו מתאים למה שהמחשב מצפה לו). הסוג השני של שגיאות ידוע בד'כ כ"שגיאות סמנטיות": יתכן שמה שתקתקת נכון מבחינה טכנית, אבל הדרך שהשתמשת בה יצרה בעיה. למשל, אם הכנסת פקודת GOSUB בתכנית אבל שכחת את הפקודה RETURN. או נתת FOR בלי NEXT, וכו'. מכל מקום, כוונותיך היו טובות, אבל האטארי אינו יכול לפעול על בסיס של כוונות בלבד ! הוא דורש דיוק.

וכאן באות לעזרתך הודעות השגיאה, מיד נביא רשימה של הודעות שגיאה, שלקחנו ישירות מן המדריך למשתמש של אטארי △ ATARI 400/800 BASIC Reference Manual

---

△ הכינוי שניתן באנגלית ע"י מתכנתים לשגיאה בתכנות הוא "באג"(bug) שפירושו חרק, שרץ, פשפש, כלומר משהו קטן, אבל מקלקל את כל העניינים. מזה נוצר הפועל to debug, לסלק את הbug. מתכנתים בארץ נתנו צורה עברית לפועל הזה, "לדבג" ומשתמשים בו בלשון היומיום שלהם בענייני תכנות כדי לציין את סילוק השגיאות ותיקון התכנית. גם אנחנו נשתמש בצורות האלה, שאמנם נכנסו לעברית "בדלת האחורית" אבל בכל זאת חיות וקיימות.

בספר רשומות 52 הודעות שגיאה, רובן עבור שגיאות קלט/פלט (input/output) לכונני דיסקטים, רשמי קסטות, מדפסות וכו'.
אבל, יש גם 17 (מס' 2-18) שגיאות המתיחסות באופן ישיר לבעיות תכנות ב BASIC. ההסברים כאן זהים לתאורים הנתנים בנספח ב' בספר BASIC Reference Manual, אלא אם כן צויין אחרת. מה שנוסיף כאן הם רמזים שיכוונו לגלוי "באגים" בתכנית שלא מופיעים כשגיאות גלויות. תהליך זה נושא לפעמים אופי של עבודת בילוש, אבל הרמזים שלהלן יעזרו לך מאד בעבודת הבילוש :

- כדאי לכתוב את התכניות במבנה מודולארי. פירוש הדבר, לנסות להשתמש בתת תכניות (GOSUB) בכל מקום שהדבר אפשרי, ביחוד כאשר צריך לחזור ולבצע אותו הדבר בחלקים שונים של התכנית.

- כדאי לשמור על עקביות בשיטה של מתן שמות למשתנים מתכנית לתכנית. הדבר גם מקל על העברת רעיונות מתכנית אחת לשניה.

- כדאי להשתמש ב REM כאמצעי לתעוד (מלשון תעודה, מסמך) התכנית. ההערות האלה אינן תופסות מקום בזכרון, ומאידך חוסכות זמן רב כשרוצים לחזור אל תכנית קודמת. כך גם קל יותר להבין את התכנית ולערוך בה שינויים.

- נוסף לשמוש במשפטי REM, כדאי לערוך רשימה של שמות המשתנים (כולל מערכים ומשתני מחרוזת) ; על הרשימה לכלול תאור מפורט של כל משתנה - שמו, תפקידו, והערות מיוחדות על מספרי שורות שבהם מתרחש שינוי חשוב בערכו של המשתנה.

הרשימה הבאה מכילה שגיאות שכיחות שודאי יופיעו בתכניותך.

קבוצה מסוימת זו של שגיאות מוסברת בפירוט רב יותר מאשר בנספח ב' של ATARI 400/800 BASIC Reference Manual.

הסיבה היא ששגיאות אלה יותר נפוצות, גם אצל אנשים כמו מחברי הספר ! כפי שציינו קודם, כולן "שגיאות ריצה" כלומר, הן לא תופענה כשגיאות עד אחרי פקודת RUN, משום שהן נכונות מבחינה תחבירית (המשפטים אולי נראים נכונים, אבל צריך להזהר !)

שגיאה מס'	תאור וסיבות אפשריות
3	שגיאה בערך. שגיאה זו עלולה לצוץ פתאום כשנמצאים במצב גרפי. הערכים של PLOT ו- DRAWTO חייבים להיות לא שליליים, כלומר, 0 או חיוביים. התכנית הבאה תגרום לשגיאה זו :

```
10 GRAPHICS 7
20 COLOR 1
30 PLOT -10,20
```

חשוב לזכור : מערכת הקואורדינטות X,Y שבשמוש הגרפיקה של אטארי איננה השיטה הקארטיזיאנית המקובלת ! פרטים נמצאים בפרקים ב' ו-ו'

5	אורך המחרוזת : השגיאה מופיעה כשמנסים להשתמש במחרוזת שהיא ארוכה מן המקום השמור עבורה במשפט DIM דוגמה :

```
5 DIM A$(5)
6 LET A$="ABCDE"
10 LET A$(6,6)="F"
```

הסבר:
התכנית תביא לשגיאה מס' 5 בשורה 10 מפני שיש רק 5 מקומות שמורים עבור A$ בשורה 5, אבל נסינו להשתמש בששה בשורה 10.
תיקון:

```
5 DIM A$(10)
6 LET A$="ABCDE"
10 LET A$(6,6)="F"
```

עצתנו היא, שאם לא בטוחים מה יהיה אורך המחרוזת, עדיף תמיד לשמור ב DIM יותר מקום ממה שנחוץ.

6	אין נתונים. השגיאה הזו מופיעה כשמשתמשים בפקודת READ לקריאת נתונים (מספרים או מחרוזות) ממשפטי DATA, אבל אין יותר נתונים שהמחשב יכול לקרוא !

השגיאה הזו נפוצה ביותר בלולאות
FOR-NEXT, כפי שנראה להלן :

```
5 DIM M(10)
10 FOR I=1 TO 10
20 READ A: M(I)=A: REM
```

אי אפשר לקרוא ישירות לתוך מערך !

```
30 NEXT I
40 DATA 1,3,5,7,9,11,13,15,17
```

הלולאה FOR-NEXT בשורות 20-30
תבצע את שורה 20 עשר פעמים,
אבל, בשורה 40 יש רק תשעה פרטי
נתונים. סיבה אפשרית אחרת
לשגיאה זו היא השמטת משפט DATA
שלא במתכוון, מתוך ששוכחים לתת
פקודת RESTORE לקריאת הנתונים
מתוך השורה המתאימה. למשל,
בתכנית הבאה :

```
10 FOR I=1 TO 5
20 READ A:PRINT A+5
30 NEXT I
40 FOR J=1 TO 5
50 READ A:PRINT J*A
60 NEXT I
70 DATA 1,3,5,7,9
```

התכנית הזו תגרום לשגיאה מס' 6
בשורה 50. אם ברצוננו להשתמש
באותו DATA, אפשר לתקן את השגיאה
ע"י הוספת שורה אחת :

```
35 RESTORE 70
```

זה יחזיר את הנתונים שצריך לקרוא
להתחלת הרשימה בשורה 70, במקום
שהתכנית תעזוב אותם אחרי קריאת
חמשה נתונים.
פשוט צריך לזכור : עבור כל משפט
READ צריך להיות משפט DATA מקביל,
ומספר הנתונים בכל משפטי ה DATA
צריך להיות מספיק עבור כל מספר של
פקודות READ.

8 משפט קלט שגוי. את השגיאה הזאת אפשר לאבחן די בקלות. פירוש הדבר הוא שערך לא מספרי נכנס (ע"י INPUT) כשצריך היה להיות מספרי. למשל :

```
10 INPUT A
20 PRINT A
RUN
?A
```

כאן, התכנית תגרום לשגיאה מס' 8 מכיון שהאות "A" איננה מספר, למרות שבתכנית היא משמשת לציון שמו של משתנה מספרי. חוק : אם מצפים שהמשתמש יתקתק משהו אחר ממספרים רצוי להשתמש במשתנה מחרוזת עבור INPUT :

```
5 DIM A$(1)
10 INPUT A$
20 PRINT A$
```

כאן לא תהיה שגיאה.

---

9 שגיאה ב DIM לגבי מערך או מחרוזת. יתכן שיש קשר לשגיאה מס' 5, או שזו שגיאה מספרית פשוטה במערך. גם למשתני מחרוזת וגם למשתני מערכים צריך לשמור מקום באמצעות הכרזות DIM. אם מנסים להתיחס לחלק של מערך או מחרוזת מעבר לגבולות גודל ה DIM , תתקבל השגיאה הזו. הנה שתי דוגמאות :

```
10 DIM A(10), B(3,3)
20 LET A(12)=2
30 LET B(3,4)=A(5)
```

```
10 INPUT NAME$
20 IF NAME$="ABE"
   THEN 10
```

בדוגמה משמאל, שני המערכים A ו-B הוכרזו ע"י DIM, אבל יש התיחסות לאברים במערך שחורגים מגבולות ההכרזה. בדוגמה מימין, יש התיחסות למשתנה NAME$ אבל לא הוכרז עליו כמחרוזת באמצעות DIM.
שגיאה זו נפוצה מאד ב BASIC של אטארי, משום שביב אטארי של BASIC דורש להכריז ע"י DIM על כל משתני

המחרוזת לפני השמוש בהם - שלא בדומה לניבים אחרים של BASIC . לכן, כאשר עומדים להשתמש במשתני מחרוזת ב BASIC של אטארי, צריך להכריז עליהם מלכתחילה ע"י DIM !

---

11 <u>נסיון לחלק באפס או התיחסות למספר</u>
<u>גדול מ- 10^{98} או קטן מ- 10^{-99}.</u>
החלוקה ב-0 איננה מוגדרת מבחינה מתמטית. וכן, מחשב אטארי אינו יכול לטפל במספרים גדולים או קטנים מאלה שציינו. אם השגיאה הזו מופיעה, עליך לבדוק את חישוביך !

---

12 <u>השורה לא במצאת</u>. שגיאה מאד נפוצה, לרוב כתוצאה מרשלנות "משרדית" פשוטה. אם משתמשים ב GOTO, GOSUB או משפטי IF-THEN, צריך לודא שמספר השורה המצויין בפקודות אלה אכן קיים. שתי דוגמאות פשוטות :

```
8 A=7
10 PRINT "HELLO"
15 IF A=5 THEN GOTO 5
20 A=A-2
25 GOTO 10
```

```
10 GOSUB 100
20 PRINT "HURRAY!"
200 LET X=30
210 RETURN
```

בדוגמה משמאל, תופיע שגיאה 12 בשורה 15 ; בדוגמה מימין תופיע השגיאה בשורה 10, משום שמספר השורה של תת-התכנית הוא 200 במקום 100.

---

13 <u>אין משפט FOR</u>. איכשהו התכנית הגיעה לפקודת NEXT אבל איננה מוצאת משפט FOR מקביל. זה קורה לעיתים תכופות ביותר בשני מקרים: 1) שיכחה פשוטה;
ו- 2) שימוש בלולאות FOR-NEXT רבות, מכוננות זו בתוך זו. נתבונן בשתי התכניות שלהלן; היכן תופיע שגיאה מס' 13?

```
10 FOR I=1 TO 10
20 PRINT I/15*22.9
30 FOR J=2 TO 10 STEP 2
40 PRINT J-I
```

```
50 NEXT I
60 NEXT J
```

בעצם, התכנית נראית בסדר, אבל היא
לא ! כשזקוקים ליותר מלולאת FOR-NEXT
אחת, הן חייבות לבוא זו בתוך זו
בשלמות ! כלומר, התכנית תוציא שגיאה
בשורה 60. כדי לתקן אותה, צריך להשלים
את הלולאה עם J לפני זו עם I.
אם נשנה:

```
50 NEXT J
60 NEXT I
```

זה יפתור את הבעיה.

---

16 שגיאה של RETURN. פירוש הדבר הוא,
שהתכנית הגיעה למשפט RETURN בטעות.
(למשל-לא היתה פקודת GOSUB שתוצאתה
היא להגיע ל- RETURN). הסיבה
הנפוצה ביותר לשגיאה זו היא שהתכנית
הגיעה ל"תת-תכנית" בעוד שהיתה צריכה
כבר להפסיק או להגמר.
הנה דוגמא :

```
10 DIM Y$(1)
20 PRINT "WOULD YOU PLAY AGAIN";
30 INPUT A$
40 IF A$="Y" THEN GOSUB 100
50 PRINT "THANKS ANYWAY."
100 PRINT "O.K., HERE WE GO AGAIN!"
110 RETURN
```

מה שיקרה כאן הוא שהתכנית תבצע קפיצה
לתת-תכנית (GOSUB) בשורה 40
כשהשחקן עונה " YES " ; אבל, אין שום
דבר בתכנית שמגן על השחקן מליפול לשורה
100 אחרי שחזר משורה 110. לכן, תבוצענה
השורות הבאות:
10, 20, 30, 40, 100, 110, 50, 100, 110.
אילו היתה התשובה " NO ", היה הרצף :
10, 20, 30, 40, 50, 100, 110.
בשני המקרים, בגלל חוסר הגנה, היתה שורה
110 מתבצעת. הפתרון כאן הוא קל ומומלץ
ביותר לכל התכניות המשתמשות ב
GOSUB/RETURN :
כדאי לתת פקודת END או STOP לפני

שמגיעים לנקודה זו (בדוגמה שלנו, זו יכולה להיות שורה 90); ואז, תמיד נפסיק לפני הבעיה שיוצרת תת-התכנית.

```
90 END
```

141 הכתב מחוץ לתחום. שגיאה זו קשורה לשגיאה מס' 3. פירושה הוא שהכתב, בין במצב רגיל של טקסט או במצבים גרפיים, איננו נמצא בתחום הנכון עבור המצב הזה. הטבלה בפרק ו' מסכמת את הגבולות האלה (נמצאת גם ב- ATARI BASIC Reference Manual)

התכניות שלפניך תגרומנה לשגיאה מס' 141:

```
10 POSITION 40, 10    10 GRAPHICS 7:COLOR 1
20 PRINT "HELLO"      20 PLOT 10,10
                      30 DRAWTO 300,200
```

בדוגמה הראשונה, בסינו למקם את הכתב מעבר לנקודת המקסימום האפקית במצב טקסט רגיל (מצב גרפי 0) - המקסימום הוא 39, כך שבקו האפקי יצאבנ אל מחוץ לתחום.
בדוגמה השניה בסינו לתת פקודת DRAWTO שהיא מחוץ לתחום במצב הגרפי מס' 7, שבו נקודות המקסימום, בהתאמה, למיקום אפקי ואנכי הן (משמאל) 95, 159.

שגיאות אחרות אם תופענה שגיאות אחרות בין שגיאה מס' 19 ושגיאה מס' 171, הן קשורות בציוד היקפי או בטיפול בקבצים. צריך ראשית לבדוק את כל הציוד ההיקפי ולראות שכל החיבורים נעשו כהלכה. שגיאה מס' 144 פירושה שמנסים לכתוב על תקליטון מוגן. רוב השגיאות האחרות בד'כ מובנות מאליהן.

נספח ב

כרטיסי BASIC של אטארי- מדריך למשתמש

כרטיסי BASIC **של אטארי** (C)1980, **לן לינדסי**

ASC

```
ASC(A$)
B=ASC("*")      B=42
```

A$ = **משתנה מחרוזת או קבוע בתוך מרכאות**

- נותן את הקוד המספרי (ATASCII) של תו מחרוזת בודד.
  לעיין גם VAL, STR$, CHR$

AND

```
A AND B
IF A>0 AND A<10 THEN GOSUB 2000
```

A = תת המשפט הראשון
B = תת המשפט השני.

- פעולה לוגית. המשפט נכון רק אם שני תת המשפטים נכונים. הערך 1 אם נכון, 0 אם לא נכון.
  לעיין גם OR, NOT

ADR

```
ADR(A$)
DIM A$(5)
C=ADR(A$)       C=2085
```

A$ = שם משתנה מחרוזת.

- נותן את כתובת האחסון ההתחלתית (במספר עשרוני) של המחרוזת המסוימת.

ABS

```
ABS(N)
A=ABS(-25)      A=25
```

N = משתנה או קבוע מספרי.

- נותן את הערך המוחלט של המספר.

ATN

ATN(N)
B=ATN(56)   B=1.55294108

N = משתנה או קבוע מספרי

● נותן את ערך הפונקציה arctangens של המספר.

BYE

BYE

● יציאה מ BASIC למערכת ההפעלה של אטארי.

(ATARI COMPUTER - MEMO PAD)

CHR$

CHR$(N)
PRINT CHR$(65)   A

N = קבוע או משתנה מספרי.

● נותן את התו שיש לו הקוד המספרי המסוים (קוד ATASCII).
לעיין גם ASC, STR$, VAL

CLOAD

CLOAD

● טוען תכנית מקסטה לזכרון ה RAM . מפעיל אוטומטית את OPEN . המחשב יזמזם פעם אחת. לחץ על PLAY במכשיר ההקלטה ולחץ RETURN

לעיין גם CSAVE, ENTER, LIST, RUN, LOAD, SAVE.

CLOG

CLOG(N)
A=CLOG(83)   A=1.91907809

N = קבוע מספרי או משתנה.

● נותן את הלוג של המספר בבסיס 10
לעיין גם LOG.

CLOSE

CLOSE #F
CLOSE #1

#F = מספר הקובץ - מספרי
קבוע או משתנה (1-7)

● סוגר קובץ F (F הוא מספר הקובץ)
לעיין גם OPEN

CLR

CLR

- מנקה את כל ערכי המשתנים ומאפס ומבטל את כל המחרוזות והמערכים המוכרזים ב- DIM
לעיין גם DIM

COLOR

COLOR N

A=3:COLOR A רגיסטר 3 עכשו שולט

N = רגיסטר של צבע (0-4)
קבוע או משתנה מספרי

בחר ברגיסטר שבו תשתמש העבודה הגרפית הבאה.

לעיין גם SETCOLOR, GRAPHICS, LOCATE PLOT, DRAWTO.

CONT

CONT

- מתחיל מחדש את התכנית בשורה שבאה מיד אחרי הפקודה שבצע כשנתנה הוראת BREAK . במצב בצוע מידי בלבד. חלקים שלא בוצעו בשורה הקודמת לא מבוצעים.

לעיין גם STOP, END

COS

COS(N)

A=COS(5) A=0.283660887

N = קבוע או משתנה מספרי.

- נותן את הקוסינוס של המספר
לעיין גם SIN.

CSAVE

CSAVE

- שומר תכנית מה RAM לקסטה. המחשב מזמזם פעמיים.עכשו לוחצים על RECORD ו PLAY במכשיר ההקלטה, ואז RETURN . מבצע OPEN באופן אוטומטי.

לעיין גם CLOAD, ENTER, LIST, RUN, LOAD, SAVE, OPEN.

DATA

```
100 DATA N
```

```
8 DATA 15,SHOES
9 READ A,B$          A=15 and B$="SHOES"
```

N = קבוע מספרי או מחרוזת.

- בשמוש רק בבצוע נדחה בתוך תכנית. כל מספר של נתונים מספריים או בתוני מחרוזת יכול לבוא אחרי DATA , מופרדים ע"י פסיקים. קריאת DATA לתכנית נעשית ע"י פקודת READ.

לעיין גם READ.

DEG

```
DEG
```

```
DEG
PRINT COS(3)        0.9986295421
```

- מורה למחשב להשתמש במעלות בכל החשובים הטריגונומטריים הבאים. ברירת המחדל היא רדיאנים.

לעיין גם RAD.

DIM

```
DIM V(B)
```

```
DIM A$(3)      or      DIM A(4,9)
```

B = מימדים מספריים של מערך או מקסימום מספר התוים במחרוזת

V = שם המשתנה המספרי או המחרוזת.

- שומר מקום למחרוזות ומערכים (חד או דו מימדיים). מותר כמה DIM בשורה, מופרדים ע"י פסיקים.

לעיין גם CLR.

DRAWTO

```
DRAWTO X,Y
```

```
PLOT 1,1
DRAWTO 5,5
```

X = קואורדינטת X של המסך

Y = קואורדינטת Y של המסך

- מצייר קו מהנקודה האחרונה שנמצא בה על המסך לנקודה המבוקשת. הקצה השמאלי העליון של המסך הוא 0,0. (GRAPHICS 3-8).

לעיין גם PLOT, POSITION, PRINT#, GRAPHICS.

END

```
END
```

● בצוע התכנית מופסק, אפשר להמשיכו ע"י .CONT

לעיין גם STOP, CONT.

ENTER

```
ENTER "T:"
ENTER "C:"
```

T = סוג אמצעי קלט/פלט.

● מכניס חלק תכנית מטייפ או דיסק, שנשמר ע"י פקודת LIST. חלק זה יתווסף באופן אוטומטי לכל תכנית קיימת.

לעיין גם LIST, CSAVE, CLOAD, SAVE, LOAD, RUN.

EXP

```
EXP(N)
PRINT EXP(3)     20.0855365
```

N = קבוע או משתנה מספרי.

● מעלה את e (2.71828179) בחזקה המבוקשת.

FOR

```
FOR L=S TO E STEP N
FOR N=15 TO 1 STEP -1.5
```

L = שם המשתנה המספרי של הלולאה.
S = הערך ההתחלתי של הלולאה.
E = הערך הסופי של הלולאה.
N = הקפיצה של (אופציה, לא הכרחי).

● הלולאה חוזרת על עצמה בכל פעם שמגיעה ל NEXT עד שעוברת את הערך הסופי של הלולאה.

לעיין גם TO, STEP, NEXT.

FRE

```
FRE(0)
PRINT FRE(0)
```

● נותן את מספר הבייטים החפשיים ב RAM שעומדים לרשות המשתמש.

GET

```
GET #F,N
OPEN #1,4,0,"K:"
GET #1,N
```

#F = מספר הקובץ (1-7)
N = שם משתנה מספרי.

- מכניס בייט יחיד מאמצעי קלט/פלט שצויין במשפט OPEN המקביל. ערך ה- ATASCII ניתן למשתנה המצויין.

לעיין גם PUT, OPEN, CLOSE, INPUT

GOSUB

```
GOSUB T
A=6:GOSUB 1000+A*10      branch to 1060
```

T = מספר שורת היעד
משתנה או קבוע מספרי.

- מסתעף למספר שורה מסוים וחוזר כשגומר (כשפוגש RETURN). אפשר להשתמש עם ON.

לעיין גם ON, RETURN, POP, GOTO.

GOTO

```
GOTO T
GOTO 250
```

T = מספר שורת היעד
משתנה או קבוע מספרי.

- ביצוע התכנית ממשיך במספר השורה המצויין. אפשר להשתמש עם ON.

לעיין גם ON, GOSUB.

GRAPHICS

```
GRAPHICS R
GRAPHICS 3     or     GR. 3
```

R = מספר המצב הגרפי (0-8)
משתנה או קבוע מספרי.

- מציין איזה מצב גרפי ישמש לתצוגת מסך. מצבים 0-2 הם לטקסט, 3-8 הם גרפיים. אפשר לקצר ל GR.

לעיין גם COLOR, SETCOLOR, PLOT.

IF

```
IF A THEN B
IF A<9 THEN PRINT "WINNER"
```

A = השואה
B = משפט, ביטוי

- מרשה בצוע משפט תנאי- מתבצע רק אם ההשוואה A נכונה. מסתמך על THEN כדי להפריד השואה A ממשפט B .

לעיין גם THEN.

INPUT

```
INPUT V
INPUT N$,P
```

V = שם משתנה מחרוזת או מספרי.

- בקשה לקלט מלוח המקשים, מופסקת ע"י לחיצה על RETURN . אפשר להכניס יותר ממשתנה אחד, מופרדים ע"י פסיקים.

לעיין גם #INPUT, GET.

INPUT#

```
INPUT #F,V
INPUT #1,A
```

#F = מספר הקובץ (1-7)
V = משתנה מספרי או מחרוזת.

- מכניס נתונים מאמצעיקלט/פלט שצויין במשפט OPEN המקביל. אפשר כמה משתנים, אם מופרדים ע"י פסיקים.

לעיין גם OPEN, GET, INPUT.

INT

```
INT(N)
PRINT INT(1.8)
```

N = קבוע או משתנה מספרי.

- נותן את המספר השלם של המספר, מעגל כלפי מטה מקצץ מה שאחרי הנקודה.

LEN

```
LEN(S$)
B$="TESTING"
A=LEN(B$)          A=7
```

S$ = משתנה או קבוע מחרוזת.

● נותן את אורך המחרוזת

לעיין גם STR$, CHR$, ASC.

LET

```
LET A=B
LET C=5*B
```

A = שם משתנה מספרי או מחרוזת.
B = קבוע או משתנה, מספרי או מחרוזת.

● המלה LET היא אופציונלית ומושמטת לעתים קרובות. משתמשים בה להצבת ערך לתוך משתנה.

LIST

```
LIST "T:", S,E
LIST 100,200
LIST "C:",20000,20099
```

"T": = סוג אמצעי קלט/פלט שממנו רוצים list
S = השורה שממנה להתחיל list
E = השורה שבה להפסיק את ה list

● נותן list, הדפס על המסך, של שורות התכנית. (אם לא מצוינות השורות, מראה את כל התכנית).

לעיין גם ENTER

LOAD

```
LOAD "T:name.ext"
LOAD "C:"
LOAD "D:TEST.05"
```

"T: = האות שמציינת את סוג אמצעי קלט/פלט
name = שם, 0-8 תוים, אופציונלי
ext = 0-3 תוים להרחבת השם.

● טוען תכנית מאמצעי קלט/פלט, כולל OPEN אוטומטי.

לעיין גם CLOAD, CSAVE, SAVE, ENTER, LIST, RUN.

LOCATE

```
LOCATE X,Y,N
LOCATE 5,8,A
```

X = קואורדינטת X ; Y= קואורדינטת Y
N = שם משתנה מספרי

- שומר את מספר רגיסטר הצבע ששולט על הנקודה המוגדרת ע"י המשתנה שצויין.

לעיין גם GRAPHICS, PLOT, COLOR, SETCOLOR, DRAWTO.

LOG

```
LOG(N)
PRINT LOG(5)      1.60943791
```

N = קבוע או משתנה מספרי.

- נותן את הלוגריתם הטבעי של מספר.

לעיין גם CLOG.

LPRINT

```
LPRINT A
LPRINT "ANSWER IS:";A
```

A = קבוע או משתנה מספרי או מחרוזת.

- דרך קיצור להוריד למדפסת. מבצע OPEN אוטומטי. ניתן להוציא להדפסה יותר מפריט אחד לביטוי כשהם מופרדים ע"י נקודה פסיק או פסיקים. (בקיצור-LP)

לעיין גם #PRINT, PRINT.

NEXT

```
NEXT
NEXT L
```

L = שם משתנה מספרי, אותו שם שמשתמשים בו ב FOR.

- סוף הלולאה FOR-NEXT. בצוע התכנית ממשיך בתחילת הלולאה עד שמתקיים התנאי של הלולאה.

לעיין גם FOR, TO, STEP.

NEW

```
NEW
```

● מוחק את התכנית ומנקה את המשתנים.

לעיין גם CLR.

NOT

```
NOT N
A=NOT Z
PRINT NOT 2
```

N = משתנה או קבוע מספרי.

● NOT הופך את ההערכה נכון/לא נכון של המשפט שבא אחריו. נכון = 1, לא נכון = 0.

לעיין גם AND, OR.

NOTE

```
NOTE #F,S,B
NOTE #1,A,B
```

#F = מספר הקובץ
S = מספר הסקטור
B = מספר הבייט בתוך הסקטור.

● זוהי פקודת DISK לציין נקודה מסוימת על הדיסק.

לעיין גם POINT, OPEN.

ON

```
ON N GOSUB L1,L2...
ON X GOTO 100,200,300
```

N = שם משתנה מספרי
L1 = מספר השורה הראשונה
L2 = מספר השורה השניה

● אפשר להשתמש ב ON יחד עם GOTO או GOSUB.
מספרי שורות היעד יכולים להיות קבועים או
משתנים, ניתן להשתמש בכמה מספרי שורה הנכנסים בתוך שורה
לוגית, (מספרי שורה אחרי המספר הראשון הם
אופציונליים).

לעיין גם GOTO, GOSUB.

OPEN

```
OPEN #F,I,P,"T:name.ext"
OPEN #1,4,0,"C:"
```

#F = מספר הקובץ (1-7)
I = קוד קלט/פלט : 4= קלט, 8=פלט, 12=שניהם.
P = קוד מדפסת (0 אם לא בשמוש)
T = אות לציון קסטה או דיסק
name =0-8 תוים לשם,אופציונלי
ext =0-3 תוים להרחבת השם

- פותח קובץ לקלט/פלט. שם ואמצעי אחסון יכולים להיות משתנה מחרוזת.

לעיין גם GET, #INPUT, PUT, #PRINT.

OR

```
A OR B
IF X>99 OR Y>99 THEN 2000
```

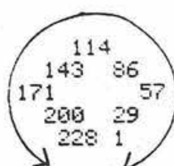
A = משפט ראשון
B = משפט שני

- הערכה לוגית, אם כל משפט נכון, ה- OR נכון. נכון = 1, לא נכון = 0.

לעיין גם AND, NOT.

PADDLE

```
PADDLE(P)
B=PADDLE(A)
```



P = מספר כפתור הכוון, קבוע או משתנה (0-7)

- הערך גדל כשמסובבים את כפתור הכוון נגד כוון השעון (1-228)

לעיין גם PTRIG, STICK, STRIG.

PEEK

```
PEEK(L)
IF PEEK(53279)=6 THEN 2000
```

L = מקום בזכרון, קבוע או משתנה מספרי.

- נותן ערך עשרוני של תוכן התא במקום המצויין בזכרון.

לעיין גם POKE, ADR.

PLOT

```
PLOT X,Y
PLOT 1,Y+5
```

X = קואורדינטת X
Y = קואורדינטת Y

● מצייר נקודה בודדת על המסך. הצבע נקבע ע"י פקודת COLOR האחרונה. הקואורדינטות יכולות להיות קבועים או משתנים מספריים. הנקודה 0,0 נמצאת בקצה השמאלי העליון.

לעיין גם DRAWTO, COLOR, GRAPHICS.

POINT

```
POINT #F,S,B
POINT #1,3,5
```

#F = מספר הקובץ
S = מספר הסקטור
B = מספר הבייט בתוך הסקטור

● זוהי פקודת DISK לציון מקום מסוים על התקליטון.

לעיין גם NOTE, OPEN.

POKE

```
POKE L,V
POKE 82,LMARG
```

L = מקום ב RAM שאליו רוצים להכניס ערך
V = ערך (0-255) שרוצים להכניס ל RAM

* שם את הערך המצויין במקום המסוים ב RAM

לעיין גם PEEK.

POP

```
POP
```

● מסלק את ה FOR-NEXT או GOSUB האחרון מן המחסנית. לשמוש להפסקת הלולאה או תת התכנית שלא באופן רגיל. הבצוע ממשיך עם המשפט הבא.

לעיין גם GOSUB, RETURN, NEXT, FOR.

POSITION

```
POSITION X,Y
POSITION 4,6
```

X = קואורדינטת X
Y = קואורדינטת Y

● קובע את הנתב במקום המצויין על המסך. הנקודה 0,0 נמצאת בקצה השמאלי העליון. מציינים קואורדינטות עם קבועים או משתנים מספריים.

לעיין גם PLOT.

PRINT

```
PRINT A
PRINT "WINNER IS:";N$
```

A =מספרי או מחרוזת

● יכולים להדפיס קבועים או משתנים. אפשר לתת יותר מפריט אחד לפקודת PRINT , מופרדים ע"י פסיקים או נקודה פסיק.
? הוא קיצור ל PRINT

לעיין גם #PRINT.

PRINT#

```
PRINT #F ; A
PRINT#6;"YOU WIN"
```

#F= מספר קובץ (1-7)
A = מספרי או מחרוזת

● מדפיס לאמצעי קלט/פלט המזוהה ע"מ משפט OPEN (#6 הוא OPEN אוטומטי לגרפיקת מסך) פריטים להדפסה יכולים להיות קבועים או משתנים.

לעיין גם PRINT, OPEN.

PTRIG

```
PTRIG(P)
10 IF PTRIG(P)=1 THEN 10
```

P =מספר כפתור כוון (0-7)
משתנה או קבוע מספרי

● נותן 1 אם הכפתור על ה paddle לא לחוץ, 0 אם לחוץ.

לעיין גם PADDLE, STICK, STRIG.

PUT

```
PUT #F,V
PUT#1,ASC("D")
```

#F= מספר קובץ (1-7)
V = קבוע או משתנה מספרי (0-255) קוד תוים של ATASCII.

- זהו ההפך של GET. שם בייט אחד המייצג את קוד ATASCII של כל תו שהוא הנשלח לקסטה או לדיסקט על פי המצויין במשפט ה-open.

לעיין גם GET, OPEN.

RAD

```
RAD
```

- מורה למחשב שכל הפונקציות הטריגונומטריות מכאן ואילך יהיו ברדיאנים (לא מעלות). ברירת המחדל היא רדיאנים.

לעיין גם DEG.

READ

```
READ A
READ N,N$
```

A = שם משתנה מספרי או מחרוזת.

- אפשר לתת ערכים ממשפטי DATA לאחד או יותר משתנים, באמצעות פקודת READ . אפשר להשתמש בכמה משתנים, מנפרדים ע"י פסיקים.

לעיין גם DATA.

REM

```
REM anything
```

- מאפשר הערות בתכנית. המחשב יתעלם מכל מה שמופיע אחרי REM וימשיך בבצוע בשורה הבאה.

RESTORE

```
RESTORE L
RESTORE 1000+G
```

L = מספר שורה - קבוע או משתנה (ברירת המחדל היא 0)

- מאפשר להשתמש שוב במשפט DATA המתחיל במספר השורה שצויין.

לעיין גם READ, DATA.

RETURN

```
RETURN
```

- חוזר מתת התכנית למספר השורה שקראה לתת התכנית עם GOSUB.

לעיין גם GOSUB, POP.

RND

```
RND(0)
X=INT(RND(0)*6)+1
```

- נותן מספר אקראי בין 0 ו- 1 (לא כולל)

RUN

```
RUN"T:name.ext"
```

"T:name.ext" אופציונלי, סוג האמצעי קלט/פלט.
RUN : מריץ את התכנית בזכרון.
RUN"C:" : טוען ומריץ תכנית על קסטה.

- מבצע את התכנית מהשורה הראשונה.
מאפס את המשתנים (אבל לא מערכים)

לעיין גם LOAD, CLOAD, SAVE, CSAVE.

SAVE

```
SAVE"T:name.ext"
SAVE"C:"
```

T = אות לציין אמצעי קלט/פלט.
name = 0-8 תוים לשם : אופציונלי.
ext = 0-3 תוים להרחבת השם

- שומר את התכנית באמצעי שצויין. מבצע OPEN באופן אוטומטי.

לעיין גם CSAVE, LOAD, CLOAD, RUN.

SGN

```
SGN(N)
IF SGN(X)=1 THEN PRINT "FINISHED"
```

N = משתנה או קבוע מספרי.

- נותן 1+ אם המספר חיובי, 0 אם הוא 0, או 1- אם הוא שלילי.

SETCOLOR

```
SETCOLOR R,C,L
```

R = מספר הרגיסטר (0-4)
C = צבע (0-15)
L = דרגות בהירות 0-14 מספרים זוגיים בלבד. הבהירות עולה עם המספר.

● שומר מידע על צבע ובהירנת באחד מחמשת הרגיסטרים של הצבע.

לעיין גם COLOR, GRAPHICS, PLOT.

SIN

```
SIN(N)
A=SIN(B)
```

N = משתנה או קבוע מספרי.

● נותן את הסינוס של המספר.

לעיין גם COS.

SOUND

```
SOUND R,P,T,V
SOUND 2,141,10,8
```

R = רגיסטר או מספר הקנל (0-3)
P = גובה הצליל (נמוך 0-255 גבוה)
V = עצמת הצליל (גבוהה 0-15 נמוכה)
T = גוון הצליל (מספרים זוגיים 0-14, צליל נקי = 10).

SQR

```
SQR(N)
PRINT SQR(9)
```

N = משתנה או קבוע מספרי.

● נותן את השורש הרבועי של המספר.

STATUS

```
STATUS
```

● נותן את הסטטוס של אמצעי קלט/פלט שנפתח.

STEP

```
FOR L=S TO E STEP N
FOR Q=1 TO 9 STEP .3
```

L = שם המשתנה המספרי של הלולאה.
S = הערך ההתחלתי של הלולאה.
E = הערך הסופי של הלולאה.
N = גודל הקפיצה (אופציונלי).

- ברירת המחדל של STEP היא 1,
משתנה הלולאה גדל בגודל ה STEP

לעיין גם TO, FOR, NEXT.

STICK

```
STICK(N)
250 IF STICK(ZJ)=15 THEN 250
```

N = מספר ידית ההגוי (0-7)

- נותן את מקומה של ידית ההגוי מספר N.
15 הוא המקום הניטרלי. מספרים אחרים
הם כמו בשרטוט.

```
 10 14 6
  \ | /
<11-15--7>
  / | \
  9 13 5
```

לעיין גם STRIG.

STOP

- הבצוע נעצר. אפשר לחדש הבצוע עם CONT
בלי שנוי מצב המשתנים.

לעיין גם END, CONT.

STRIG

```
STRIG(N)
IF STRIG(1)=0 THEN PRINT "FIRE!"
```

N = מספר ידית ההגוי (0-7)
משתנה או קבוע מספרי.

- נותן 1 אם הכפתור האדום של ידית ההגוי
מספר N לא לחוצה, ו-0 אם לחוצה.

לעיין גם PTRIG, STICK.

STR$

```
STR$(N)
N$=STR$(N)
```

N = קבוע או משתנה מספרי.

● הופך מספר למחרוזת הבניה מהספרנת של המספר. המספר 154 הופך ל "154".

לעיין גם CHR$, VAL, ASC.

THEN

```
IF A THEN B
IF A=10 THEN PRINT "WINNER"
```

A = השוואה
B = משפט

● אם ההשוואה נכונה אזי מבצע את המשפט שאחרי THEN . יכול לבוא עוד IF THEN.

לעיין גם IF.

TO

```
FOR L.=S TO E STEP N
FOR A=1 TO 30
```

L = שם משתנה מספרי של הלולאה.
S = ערך התחלתי של הלולאה.
E = ערך סופי של הלולאה.
N = גודל הקפיצה (אופצינלי).

● TO מפריד בין הערך ההתחלתי והערך הסופי של משתנה הלולאה.

לעיין גם STEP, FOR, NEXT.

TRAP

```
TRAP L
250 TRAP 250 : INPUT N
```

L = מספר שורה, משתנה או קבוע.

● אם יש שגיאה, ה TRAP האחרון שולט והבצוע ממשיך בשורה שציין.
אחרי כל שמוש צריך להחילו מחדש.
אפשר להחיל TRAP כמה פעמים שרוצים.

USR

```
USR(V)
USR(15)
```

V = ערכי קלט (לפחות אחד).

- מבצע תכנית בשפת מכונה תוך שמוש בערכי הקלט שצויינו.

VAL

```
VAL(S$)
N=VAL(N$)
```

S$ = קבוע או משתנה מחרוזת.

- הופך את קריאת S$ הרגילה משמאל לימין לערך המספרי של המספר (עד שפוגש בתו לא מספרי). אם לא מוצא תוים מספריים מודיע על שגיאה.

לעיין גם STR$, ASC, CHR$.

XIO

```
XIO C,#F,0,0,"T:"
XIO 18,#6,0,0,"S:"
```

C = מספר הפקודה
#F = מספר הקובץ
T = האות לציון אמצעי קלט/פלט

- אפשר להשתמש בזה בפעולות דיסק וגרפיקה. במצב גרפי 2 משתמשים בנתוני דמה 0,0. הדוגמה ממלאת שטח במסך בין נקודות עם הצבע שנקבע ע"י POKE 765,C, כאשר C=מספר הצבע.

לעיין גם POKE, PLOT, DRAWTO.

מפתח אמצעי קלט/פלט		משתמש במילות המפתח
"S:"	מסך	ENTER "T:שם האופציה"
"E:"	עורך	LIST "T:",התחלה,סוף
"K:"	לוח מקשים	LOAD "T:שם האופציה"
"C:"	קסטה	OPEN#N,I,P,"T:שם"
"P:"	מדפסת	RUN"T:שם האופציה"
"D:"	דיסק	SAVE"T:שם האופציה"
		XIO C,#F,0,0,"T:"

החלף את ה T במשפט באות המתאימה של אמצעי הקלט/פלט הרצוי. למשל RUN"C:"

???	הכי גבוה	יותר גבוה	גבוה	בינוני	נמוך	הערה
4	8	16	32	64	128	B
4	8	17	34	68	136	Bb
5	9	18	36	72	144	A
5	9	19	38	76	153	G#
5	10	20	40	81	162	G
6	11	21	43	86	172	F#
6	11	23	45	91	182	F
6	12	24	48	96	193	E
7	13	25	51	102	204	D#
7	13	27	54	108	216	D
7	14	28	57	114	229	C#
8	15	30	60	121	243	C

1. מספר השורה צריך להיות בין 0 ו- 32767.
2. שוליים שמאליים : POKE 82,X ; ימניים : POKE 83,X
3. מצבים גרפיים 3-8 פותחים את המסך כקובץ מס' 6.
4. A$(LEN(A$+1)=B$: כדי לקבל A$=A$+B$.
5. תת מחרוזות - A$(אות ראשונה, אות אחרונה)
6. 9 מחכה למקש
7. עבור 4 מקשי בקרה מיוחדים ; OPTION=3, SELECT=5, START=6, 7= אף אחד לא לחוץ
8. כדי לעצור/להתחיל מחדש תצוגת מסך, ללחוץ CTRL ו-1.
9. מנוע רשם קסטות : POKE 54018,X ; 52=ON, 60 = OFF.
10. ניתוק נתב : POKE 752,X OFF=1, ON=0
11. לוגי : 1=נכון, 0=לא נכון (בדרך כלל)
12. ATASCII CAPS+32 = אותיות קטנות + 128 = אותיות הפוכות (רקע/תו)
13. PEEK(195) נותן את מספר השגיאה.
14. הגנת טלביזיה - POKE 77,X, OFF=0, ON=128.

צבע	C#	צבע	C#
כחול	8	אפור	0
כחול בהיר	9	זהב	1
טורקיז	10	כתום	2
ירוק-כחול	11	כתום אדום	3
ירוק	12	ורוד	4
צהוב-ירוק	13	סגול	5
ירוק-כתום	14	כחול סגול	6
כתום בהיר	15	כחול	7

טקסט 1 ו-2	צבע	ברירת מחדל
CAPS	זהב	SETCOLOR 0,2,8
LOWER	ירוק בהיר	SETCOLOR 1,12,10
RVS CAPS	כחול כהה	SETCOLOR 2,8,12
RVS LOWER	אדום	SETCOLOR 3,4,6
רקע	שחור	SETCOLOR 4,0,0

מס' מצב גרפי

מס' ה SETCOLOR	3	4	5	6	7	8
COLOR 1	0	0	0	0	0	1
COLOR 2	1	4	1	4	1	2
COLOR 3	2	0	2	0	2	1
COLOR 4	4	4	4	4	4	2

מס' הצבע בשלט ע"י

SETCOLOR0	1	1&3	1	1&3	1	-
SETCOLOR1	2	-	2	-	2	1&3
SETCOLOR2	3	-	3	-	3	2&4
SETCOLOR3	-	-	-	-	-	-
SETCOLOR4	4	4	4	4	4	-

מצב גרפי	עמודה	פיצול/שורה	בייטים
0 txt	40	24 / -	993
1 txt	20	24 / 20	513
2 txt	20	12 / 10	261
3 grp	40	24 / 20	273
4 grp	80	48 / 40	537
5 grp	80	48 / 40	1017
6 grp	160	96 / 80	2025
7 grp	160	96 / 80	3945
8 grp	320	192 / 160	7900

RVS מתיחס להיפוך רקע/תו.

נספח 3

פקודות POKE של אטארי

POKE היא מלה שמורה ב BASIC ברוב המיקרו מחשבים, כולל המחשב הביתי של אטארי. יחד עם זאת, נרתעים ממנה אנשים רבים, משום שהיא משפיעה על המחשב ישירות ובאופן מידי.

אולם אנו נפרט ונסביר כמה פקודות POKE שעשויות לעניין אותך, ובשמוש נכון הן תהיינה לעזר רב. יש פקודות POKE רבות אחרות בנספח א' של

Atari 400/800 BASIC Reference Manual

POKE 77,x

POKE 77,128 - מציב את המחשב ב ATTRACT MODE. כל דבר הנמצא על המסך במשך זמן ארוך עלול לשרוף כמה מזרחני הצבע המרכיבים את המסך. אם לא תקתקת על המקשים במשך זמן מסוים, יקפוץ מחשב אטארי ל- ATTRACT MODE, וישנה את הצבעים מבהירים לכהים וחוזר חלילה כדי למנוע את שריפת המסך.
המקום 77 בזכרון עוקב אחרי הזמן שעבר מן הפעם האחרונה שתקתקת על המקשים. בכל פעם שנוגעים בלוח המקשים, התא מס' 77 מתאפס מחדש. הוא מגדיל את ערכו מדי כמה שניות. כשהוא מגיע ל 128 הוא קופץ ל ATTRACT MODE. כך, אם מכניסים את הערך 128 (ע"י POKE) לתא מס' 77, המחשב "חושב" שהגיע הזמן להגן על המסך, וקופץ למצב ATTRACT.

POKE 77,0 - זה ההפך מלהכניס את הערך 128 למקום 77. אם נשים שם 0 יתחיל המחשב לספור מחדש עד למצב ATTRACT.
פקודה זו שמושית כשאטארי מבצע דמויות הנפשה (או "סרטים מצויירים") המסך מתחלף, כך שאין צורך לדאוג שמא ישרף. כך, כדי למנוע התחלה של מצב ATTRACT, כדאי להכניס פקודת POKE זו בתכנית לעתים תכופות.

POKE 755,x

המקום 755 עוקב אחר הדרך שבה מנצגים התוים על המסך. באופן רגיל ערכו 2. אם נכניס שם 4 ינצגו כל התוים הפוכים על ראשם. POKE 755,4 יהפוך את התוים על ראשם. POKE 755,2 יחזיר אותם על רגליהם.

POKE 764,x

מקום זה עוקב אחרי המקש האחרון שתוקתק, עד שהמחשב משתמש בו. לעתים מתיחסים לכך כאל ה"חוצץ" של לוח המקשים. המחשב תמיד יזכור את המקש האחרון שהיה בשמוש. אם נתקתק שני מקשים לפני שמחשב משתמש בהם, המחשב ישכח את המקש הראשון. POKE 764 ,255 ינקה את החוצץ, גם אם הקשת במקש מסוים. הפקודה שמושית ממש לפני פקודת INPUT, משום שהיא נותנת התחלה נקיה. למשל, בתכנית הבאה :

```
10 PRINT "HIT A KEY"
20 FOR LOOP=1 TO 999 : NEXT LOOP
30 INPUT X
```

בהרצת התכנית יקרה הדבר הבא : על המסך יופיע המשפט "HIT A KEY", ועלינו ללחוץ מיד על מספר כלשהו (X הוא משתנה מספרי), בלי ללחוץ RETURN. המספר הזה מאוחסן כעת במקום 764, ועם הופעת סימן השאלה (שהושהה ע"י הלולאה בשורה 20), יופיע המספר יחד אתו. הפקודה:

```
25 POKE 764,255 :REM CLEAR KEYBOARD BUFFER
```

תמנע את התופעה, והמחשב שנכח את המספר שתקתקת לפני הדפסת סימן השאלה לבקשת INPUT.

הפקודה PEEK 764 נותנת את המקש האחרון שבו השתמשו. למשל,

```
100 IF PEEK(764)=255 THEN 100 :REM WAIT TILL KEY IS HIT
```

השורה הזו תחכה עד שיתקתקו על מקש מסוים, לפני המשך בצוע התכנית.

POKE 83,x · POKE 82,x

שני המקומות האלה שולטים על השולים של המסך. השולים משמאל הם בשליטת 82 והימניים בשליטת 83. אפשר לשנות את השולים בכל עת ע"י פקודת POKE.

כדי למצוא מה השולים כרגע, בתקתק :

```
PRINT PEEK(82), PEEK(83)
```

באופן רגיל הערכים הם 2 עבור השולים השמאליים ו-39 עבור הימניים. זה נותן לך שורה באורך של 38 תוים, אבל אפשר להכניס 40 תוים במסך, אם רוצים, ע"י הפקודה:

```
POKE 82, 0
```

וכך נקבל אורך שורה מירבי. האם שמת לב שהמלה READY הוזזה בשני תוים שמאלה ?

וכעת:

```
POKE 83, 10
```

הכל עדיין בראה אותו דבר. אבל תנך תקתוק (למשל שורה שלמה של K), רואים שבפחנת מחצי המסך התוים קופצים לשורה הבאה. האם זה מפתיע ? עם ניקוי המסך (לא ע"י (SYSTEM RESET), אלא ע"י(CLEAR) (SHIFT)), השוליים עדיין שונים. הם ישארו שונים עד פקודת POKE אחרת ל-82 או 83, או עד לחיצה על כפתור RESET.

נספח 4

שעונים, זמן ותזמון קלט

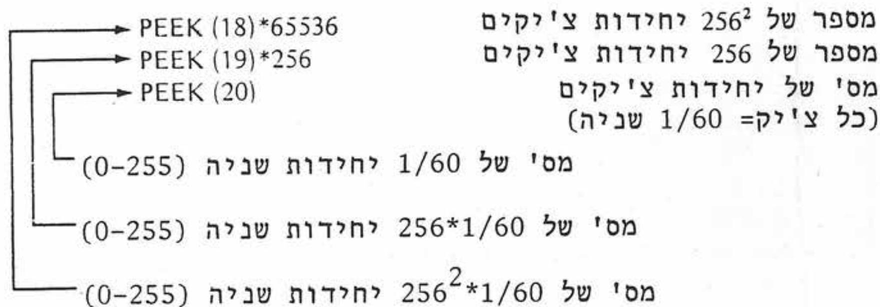
למחשב אטארי יש שעון הבנוי בתוכו, שמאפשר לך לכתוב תכניות שדורשות מעקב אחרי זמן. הדבר שמושי מאד כשרוצים להקצות זמן לתגובתו של שחקן. הנה תאור קצר של השעון "צ׳יק" של אטארי :

שלשה בייטים של זכרון מודדים את ה"זמן" ע"י כך שהם שומרים את מספר המחזורים שבשמוש יחידת העבוד המרכזית 6502 של אטארי. המחזורים מתרחשים בקצב של 1/60 של שניה. משך הזמן נשמר בתאי הזכרון מס׳ 18, 19, 20, החל מרגע הדלקת המחשב ועד ניתוקו. מתכנתים קוראים למשך הזמן 1/60 של שניה בשם האנגלי "Jiffy", שפירושו מהר מאד, בן רגע, ואנחנו נכנה אותו "צ׳יק" - ומכן שמו של ה"שעון" לשיטה זו של מדידת הזמן.

כל אחד ממקומות אלה בזכרון פועל כחלק מן השעון ; במקום למדוד את הזמן בשעות, דקות ושניות, כל תא כזה עוקב ושומר על מספר "צ׳יקים" מ-0 עד 255. כשהתא הראשון מתמלא הוא גולש לתא הבא, וכן הלאה. זה דומה מאד לשעון דיגיטלי, רק שהשעון של אטארי יודע לספור רק ב"צ׳יקים" במקום שניות, לכן, עלינו לעשות קצת חישובים כדי שיוכל לשמש אותנו.

הבה נתבונן בדוגמה. כדי שהשעון יתחיל ללכת בתכנית כלשהי, דרושה לנו פקודה כזאת :

```
10 LET TIME1 = INT((PEEK(18)*65536) + (PEEK(19)(*256) +
   PEEK (20));
```



זה יתחיל את פעולת השעון כך שתהיה לנו "נקודת זינוק".
אבל, המספר הזה ינתן לנו ב"צ'יקים" במקום בשניות.
כדי להפוך אותו לשניות, עלינו לחלק ב-60 (כל צ'יק הוא
1/60 של שניה). לכן, נוכל להוסיף :

```
20 TIME1 = INT(TIME1/60)
```

או, יכולנו לכתוב את השורה המקורית כך :

```
10 TIME1 = INT(((PEEK(18)*65536)+(PEEK(19)*256)+
   PEEK(20))/60)
```

כדי לראות מהו הזמן ההתחלתי, אנו פשוט מדפיסים את
ערכו :

```
30 PRINT "START TIME = ";TIME1
```

וכעת, אם ברצוננו לקבוע בכל רגע נתון כמה זמן
עבר, עלינו רק להפחית את הזמן ההתחלתי (TIME 1)
מהזמן הנוכחי :

```
6000 REM – SUBROUTINE TO GET ELAPSED TIME –
6010 NOW = INT(((PEEK(18)*65536)+(PEEK(19)*256)+
     PEEK(20))/60)
6020 TIMEGONE = NOW - TIME1
6030 RETURN
```

אם נדפיס את TIMEGONE, נמצא כמה שניות חלפו מרגע
שהשעון התחיל לפעול. (חשוב לזכור : מוכרחים לכוון
את השעון בכל תכנית שרוצים שתכיל מדדי זמן ; האטארי
תמיד ימנה "צ'יקים"; אבל לא יהיה באפשרותנו לדעת אותם
אלא אם כן נכוון את מהלך השעון כפי הדרוש לנו).

נוכל להשתמש בתת התכנית הזו כדי לדעת כמה זמן לקח
למישהו לענות על שאלה בחשבון באופן הבא :

```
100 TIME1=INT(((PEEK(18)*65536)+(PEEK(19)*256)+
    PEEK(20))/60)
110 PRINT "WHAT'S 4*7 ";
120 INPUT X
130 GOSUB 6000
140 PRINT "IT TOOK YOU ";TIMEGONE;"SECONDS TO
    ANSWER THAT."
```

מאחר שיש סכוי לשמוש די תכוף בתכבית הזו, זה רעיון טוב לעשות מ"מזניק השעון" הזה תת תכבית כשלעצמה. אפשר לעשות זאת ע"י מספור מחדש של שורה 100 (או שורה 10 בדוגמה הראשונה) ולהוסיף RETURN בסוף תת התכבית. ואז, כדי "להזניק" את השעון, צריך רק לתת GOSUB בשורה הזו :

```
100 GOSUB 5900:REM - השעון מתחיל לרוץ
...
...
130 GOSUB 6000:REM - מקבלים את הזמן שחלף
...
...
5900 REM - - SUBROUTINE TO START THE CLOCK - -
5910 TIME1 = INT(((PEEK(18)*65536)+(PEEK(19)*256)+
     PEEK(20))/60)
5999 RETURN
6000 REM - - SUBROUTINE TO GET ELAPSED TIME - -
6010 NOW = INT(((PEEK(18)*65536)+(PEEK(19)*256)+
     PEEK(20))/60)
6020 TIMEGONE = NOW - TIME1
6030 RETURN
```

נספח 5

לוח תוצאות:
תת תכנית ליותר משחקן אחד

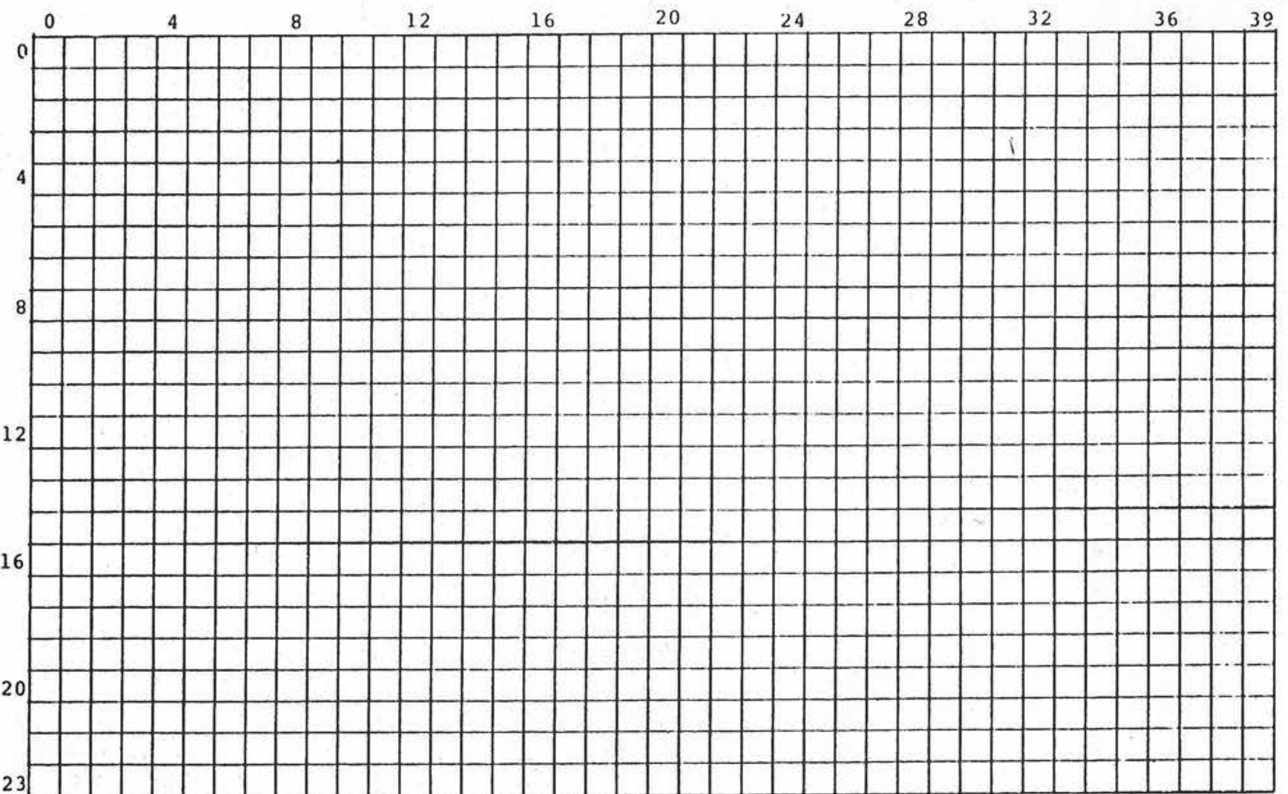
```
10 REM–SCOREKEEPER FOR MORE THAN ONE PLAYER
15 REM–TED M. KAHN
20 PRINT "HOW MANY PLAYERS";
25 INPUT P
30 DIM S(P)
35 FOR I=1 TO P:LET S(P)=0:NEXT I
100 REM–TO CALL SCOREKEEPER
101 REM LET W=NUMBER OF WINNER
102 REM (E.G., PLAYER 1,2,3, . . . ETC.)
105 REM–THEN GOSUB 6000
6000 REM–ADD TO WINNER'S SCORE
6005 REM–AND SHOW TOTALS
6010 S(W)=S(W)+1
6020 GRAPHICS 1:REM SHOW SCORES
6030 FOR I=1 TO P
6040 PRINT #6;"PLAYER;I;"=";S(I)"
6050 PRINT #6
6060 NEXT I
6099 RETURN
```

נספח 6

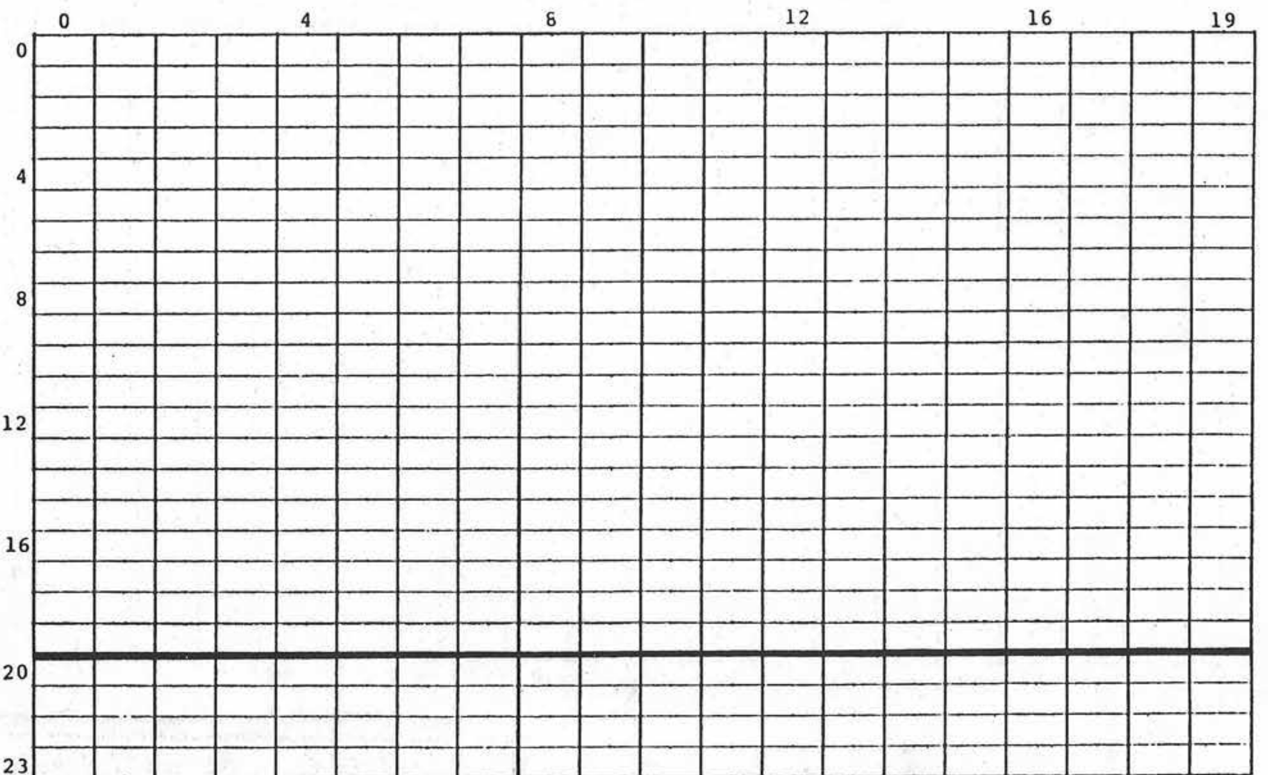
דפי שרטוט למסך
למצבים הגרפיים 8-0

ברצוננו להודות למר ביל קאריס ממחלקת המחשבים האישיים של אטארי, בע"מ על הרשות שנתן להדפיס את הדפים הללו. מר קאריס פתח אותם בתקופת עבודתו כמנהל השרותים הטכניים של אטארי.

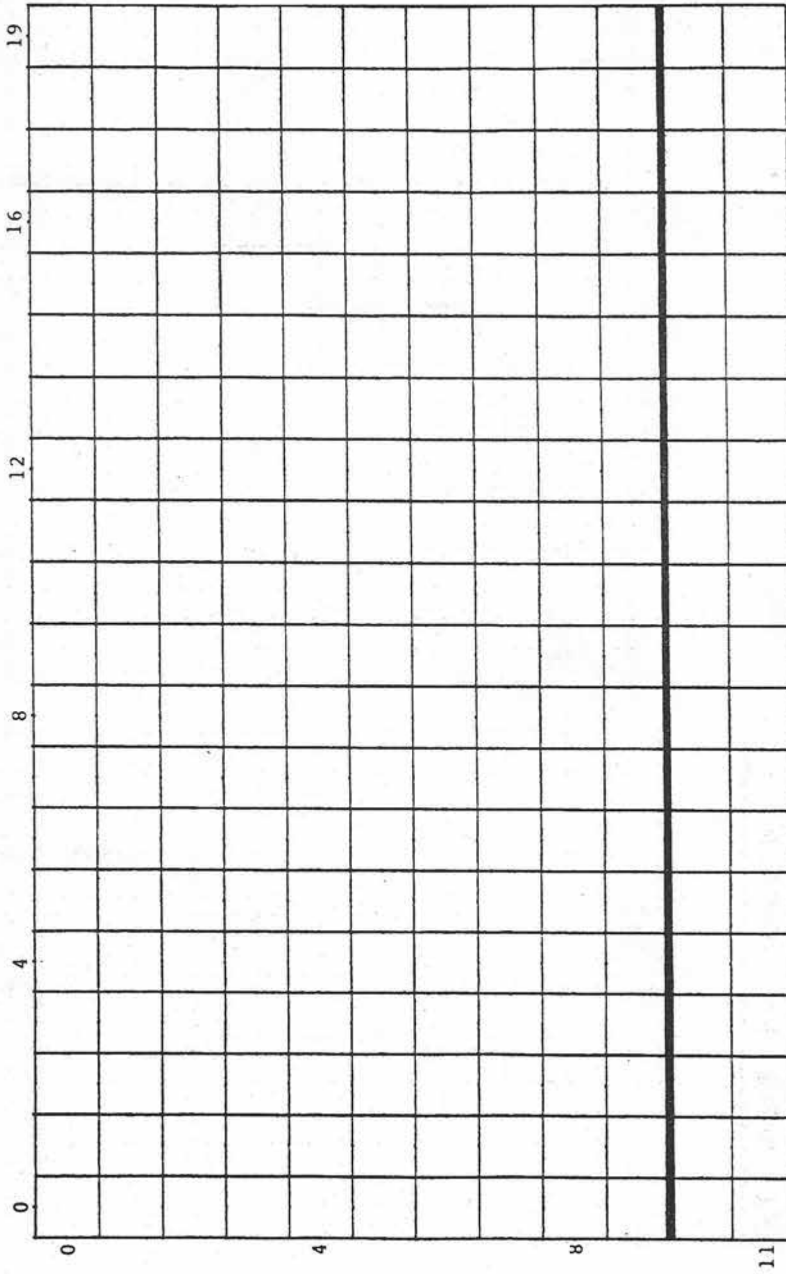
Graphics Mode 0



Graphics Mode 1

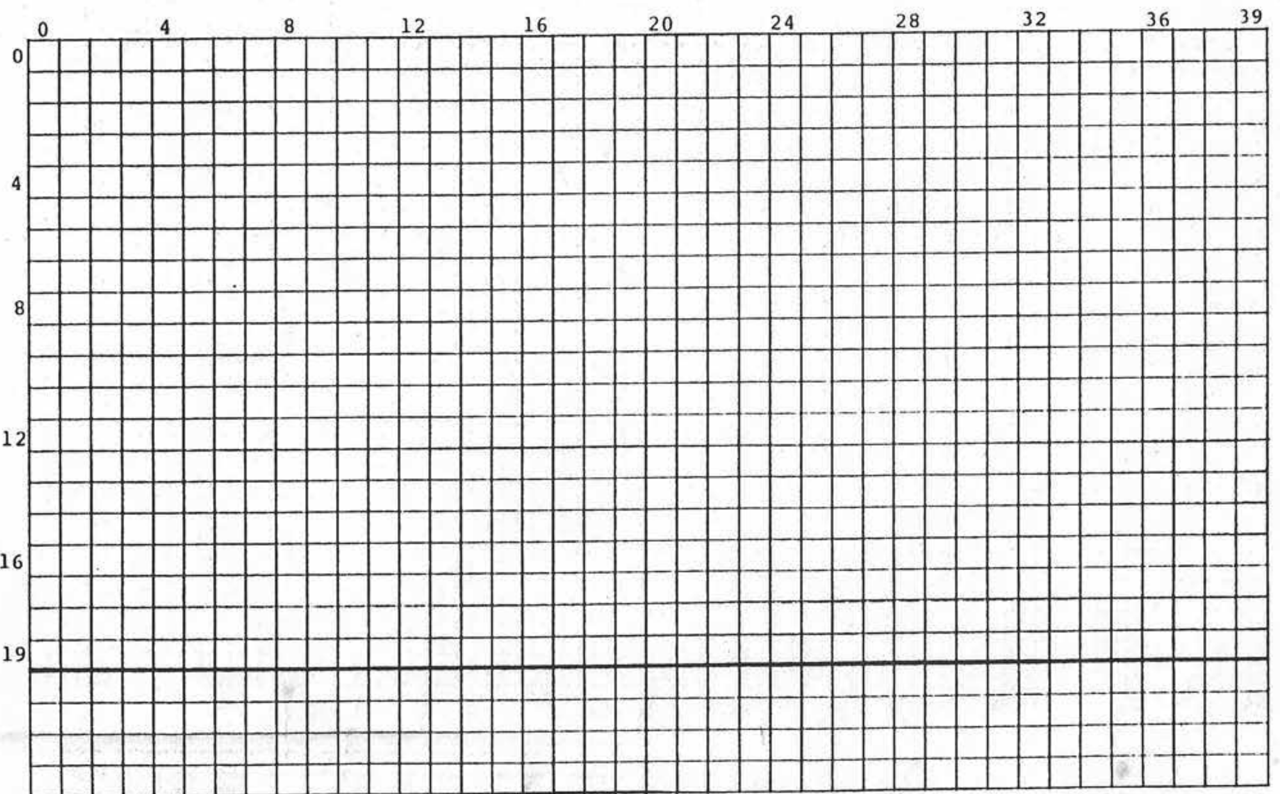


Graphics Mode 2



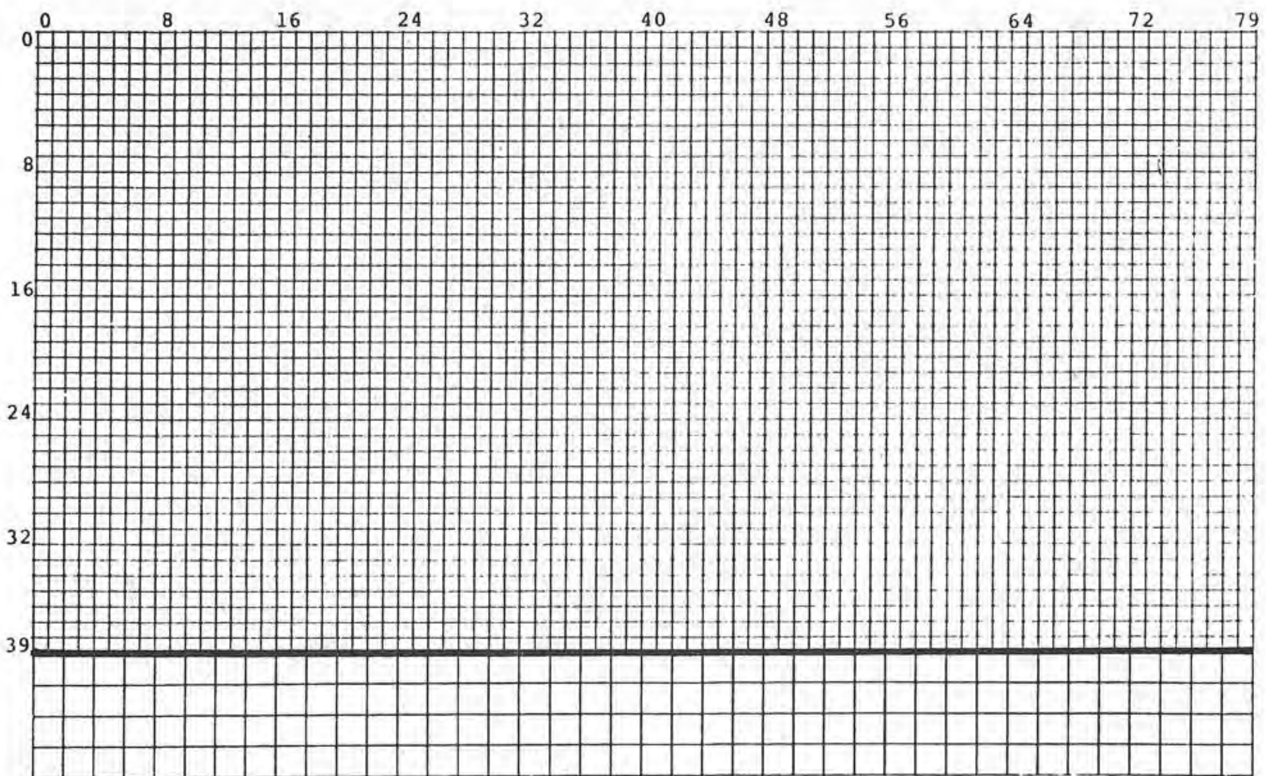
Graphics Mode 3

עם חלון טקסט

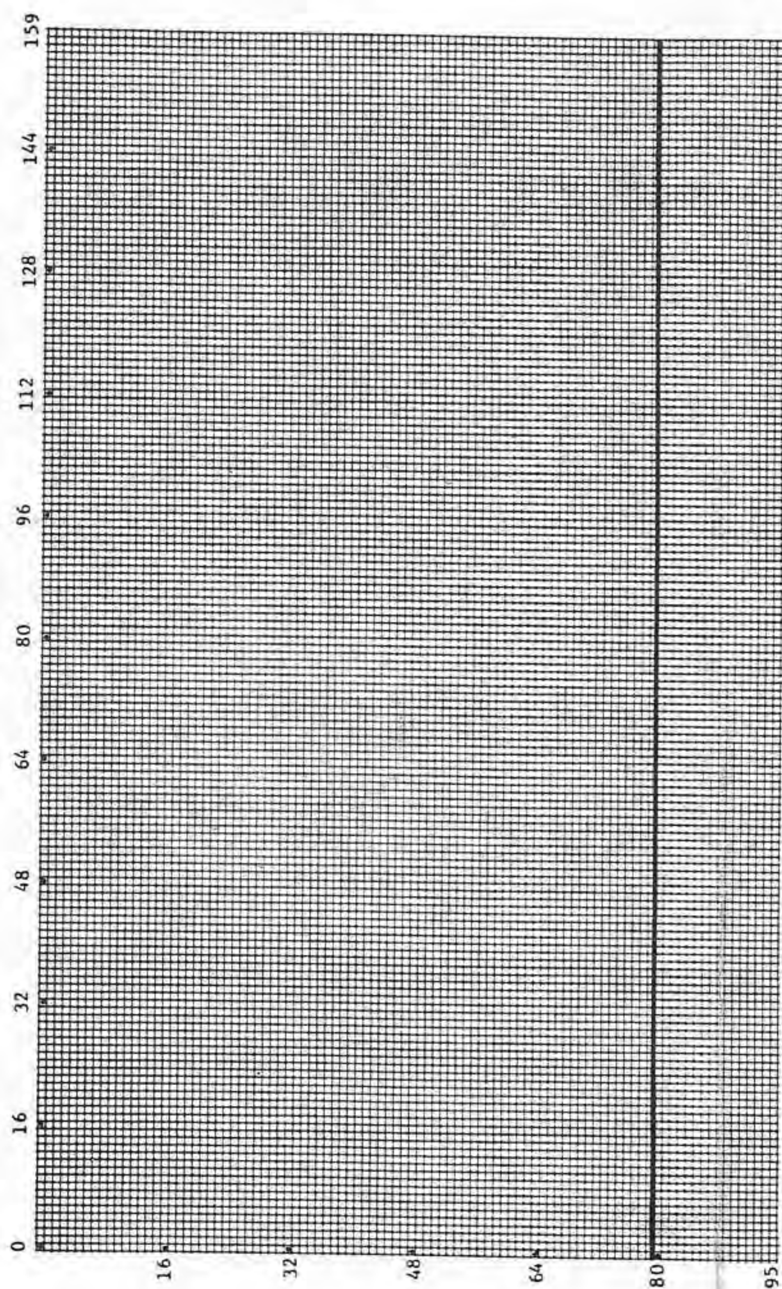


Graphics Mode 4 or 5

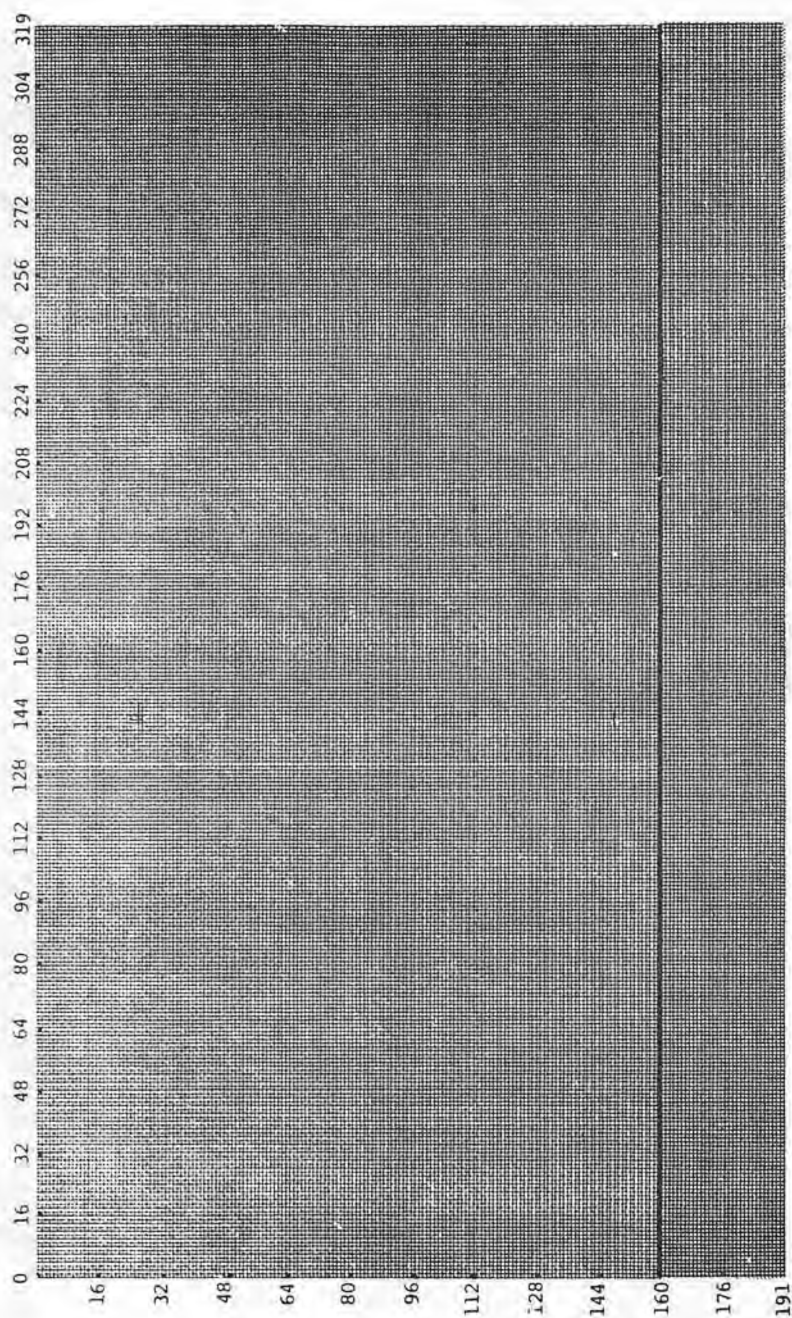
עם חלון טקסט



Graphics Mode 6 or 7



Graphics Mode 8



נספח 7

קוד ATASCII של התווים

הטבלה הבאה תוכננה כדי להקל על המעבר מ-ATASCII למחרוזות תוים. יתכן שידועים לך המקשים שבהם צריכים להשתמש בתכנית, אבל יתכן שלא ידועים לך ערכי ה- ATASCII שלהם. טבלת ה- ATASCII של לוח המקשים תציל אותך ! פשוט מוצאים את המקש בטבלה, ומקבלים את כל ערכי ATASCII האפשריים על המקש עצמו. שימו לב, שלכל אחת מ-26 האותיות יש 6 ערכים אפשריים, תלוי במצב : SHIFT, INVERSE או CONTROL . הפינה השמאלית התחתונה של הטבלה מראה כיצד לפרש ולהבין כל אחד מן המספרים הללו.

וכעת, נניח שרוצים שהמשתמש בתכנית ילחץ על Y עבור המילה YES , N עבור NO , או H עבור HELP (עזרה, "הצילו" - דוקא נחמד להכניס אופציה כזו לתכנית). אבל לא מעניין אותך אם ה- Y הזו היא אות גדולה או קטנה, או אפילו אם היא במצב INVERSE או CONTROL . פשוט מעניין אותך לדעת אם לחצו על אחד משלשת המקשים האלה (ולהתעלם מכל השאר). אז זה מה שעושים : מחפשים קודם בטבלה את המקשים, ורושמים את הערכים שלהם. בדוגמה הזו נכתוב :

Y = 89, 121, 25, 217, 249, 153
N = 78, 110, 14, 206, 238, 142
H = 72, 104, 8, 200, 232, 136

וכעת פשוט כותבים תת תכנית אחת כדי להשיג לחיצה על המקש, ותת תכנית נוספת כדי לבדוק אם זהו אחד מהערכים שלהם מצפים. אם הערך איבו בר תוקף, חוזרים ומבקשים לחיצה נוספת. אם המספר נכון, מחזירים אותו כאות Y גדולה, N או H. יתכן שזה בשמע מורכב ומסובך, אבל בעצם זה פשוט למדי. לפנינו חלק מתכנית לדוגמה, כולל שתי תת התכניות שהזכרנו :

ערכים עבור: CHR$(V)- GET # 1, V-POKE X, V-PEEK (X)- ASC ("X")

— @ 64 [192] 8 56 [184]	— (40 [168] 9 57 [185]	—) 41 [169] Ø 48 [176]	CLEAR 125 CLEAR 125 < 60 [188]	CHAR INST 255 LINE INST 157 > 62 [190]	CHAR DELETE 254 LINE DELETE 156 BACK S 126	BREAK	SYSTEM RESET

9 [137] i 105 [233] I 73 [201]	15 [143] o 111 [239] O 79 [207]	♣ 16 [144] p 112 [240] P 80 [208]	↑ 28 - 95 [223] - 45 [173]	↓ 29 | 124 [252] = 61 [189]	RETURN 155	OPTION PEEK (53279)=3

11 [139] k 107 [235] K 75 [203]	12 [140] l 108 [236] L 76 [204]	♠ 123 [251] : 58 [186] ; 59 [187]	← 30 \ 92 [220] + 43 [171]	→ 31 ^ 94 [222] * 42 [170]	CAPS LOWR	SELECT PEEK (53279)=5

13 [141] m 109 [237] M 77 [205]	♥ 0 [128] [91 [219] , 44 [172]	♦ 96 [224]] 93 [221] . 46 [174]	— ? 63 [191] / 47 [175]	RVS OFF RVS ON	SHIFT	START PEEK (53279)=6

הטבלה תוכננה ע"י לן לינדסי

SPACE 32 — ATASCII CHARACTER CODES — REVERSE SPACE 160

```
100 PRINT "YOUR CHOICE (Y,N, OR H):";
110 GOSUB 1000 : REM GET A VALID RESPONSE
120 IF C$="H" THEN GOSUB XXXX : REM ROUTINE
    FOR HELP
130 IF C$="Y" THEN GOSUB YYYY : REM ROUTINE
    FOR YES
140 IF C$="N" THEN GOSUB ZZZZ : REM ROUTINE
    FOR NO . . . [PROGRAM HERE]
999 END
1000 C$=" " : REM ROUTINE TO GET Y, N, OR H
1010 GOSUB 2000 : REM GET THE CODE OF THE KEY
     PRESSED
1020 IF CODE=89 OR CODE=121 OR CODE=25 OR
     CODE=217 OR CODE=249 OR CODE=153 THEN C$="Y"
1030 IF CODE=78 OR CODE=110 OR CODE=14 OR
     CODE=206 OR CODE=238 OR CODE=142 THEN C$="N"
1040 IF CODE=72 OR CODE=104 OR CODE=8 OR
     CODE=200 OR CODE=232 OR CODE=136 THEN C$="H"
1050 REM IF C$ STILL = " " THEN IT WAS NOT VALID
1060 IF C$=" " THEN 1010 : REM GET ANOTHER KEY
1099 RETURN : REM C$ NOW IS EITHER Y, N, OR H
2000 OPEN#1,4,0,"K:" : REM THIS COULD BE AT THE
     PROGRAMS BEGINNING
2010 GET#1,CODE: REM GET A KEY
2020 CLOSE#1 : REM NOT NEEDED IF LINE 2000 IS AT
     PROGRAMS BEGINNING
2099 RETURN : REM CODE IS THE VALUE OF THE KEY HIT
```